

Universidad Politécnica de Madrid

Escuela Universitaria de Ingeniería Técnica de Telecomunicación



Proyecto Fin de Curso (Plan 2000)

Desarrollo de Herramienta Software para el estudio del complejo
QRS en las señales de ECG

Autor: Felipe Herranz Rabanal

Tutor: Luis Arriero Encinas

Septiembre de 2013



PROYECTO FIN DE CARRERA PLAN 2000

E.U.I.T. TELECOMUNICACIÓN

TEMA: Procesado Digital de la señal

TÍTULO: Desarrollo de herramienta software para el estudio del complejo QRS en las señales de ECG

AUTOR: D.Felipe Herranz Rabanal

TUTOR: D.Luis Arriero Encinas

Vº Bº.

DEPARTAMENTO: ICS

Miembros del Tribunal Calificador:

PRESIDENTE: D.Gerardo Pérez Villalón

VOCAL: D.Luis Arriero Encinas

VOCAL SECRETARIO: D.David Osés del Campo

DIRECTOR:

Fecha de lectura:

Calificación:

El Secretario,

RESUMEN DEL PROYECTO:

Las enfermedades cardíacas son un problema grave y en aumento en la sociedad actual. El estudio automático del complejo QRS es fundamental en el diagnóstico de enfermedades cardíacas. Para ayudar al diagnóstico se propone una herramienta software que analiza el complejo QRS en señales de electrocardiograma.

Los pasos a seguir para diseñar dicha herramienta serán:

- Estudio de los diferentes métodos de detección del complejo QRS.
- Propuesta de una herramienta de detección del complejo QRS en un electrocardiograma.
- Diseño de un paquete software que realice diferentes estudios del complejo QRS con sus "interfaces" gráficas para facilitar la aplicación de la herramienta propuesta a nivel usuario.

Agradecimientos

A mis padres, Felipe y María Ángeles, y también a mi hermano Carlos por su amor, cariño y apoyo incondicional.

A mis amigos por los buenos tiempos pasados juntos.

Por último, quisiera agradecer a mi tutor de PFC, Luis Arriero Encinas, por su dedicación y apoyo para la correcta realización de este proyecto.

A todos ellos les doy las gracias.

Índice

Contenido

Índice.....	3
Índice de Figuras y Tablas.....	5
Introducción	8
1. Principios Básicos del Electrocardiograma	12
1.1. El proceso celular subyacente en el ECG	12
1.2. Física Básica de la electrocardiografía	14
1.3 La señal de electrocardiograma.....	17
1.4 Patrones anormales en la señal ECG.....	21
1.5 Arritmias Cardíacas.....	24
1.6 Otras anormalidades: isquemia	28
1.7 Introducción al análisis de la señal ECG	30
2. Características de la señal ECG desde el punto de vista de procesamiento digital de la señal.....	31
2.1 Análisis espectral de la señal ECG.....	31
2.2 Características clínicas del ECG.....	34
2.3 Detección de arritmias.....	38
2.4 Ruido y artifacts en la señal ECG.....	39
3. Descripción de los algoritmos	41
3.1 Filtro paso alto para eliminar Baseline Wander	41
3.2 Filtro paso bajo para extraer el Baseline Wander.....	42
3.3 Filtros Notch 50/60 Hz.....	43
3.4 Filtro adaptativo 50/60 Hz.....	44
3.5 Nociones comunes sobre los algoritmos de detección QRS:	45
3.6 Detector QRS con etapas de filtrado lineal y no lineal	46
3.7 Detector QRS con preprocesador basado en filtrado paso banda y transformada Hilbert.....	51
4. Comparativa entre los diferentes algoritmos	53
4.1 Tasa de aciertos	53
4.2 Eficiencia computacional.....	54
4.3 Sumario.....	56

5. Descripción de la herramienta software.....	57
5.1 Menú Principal	57
5.2 Desarrollo de nuevos métodos.....	58
5.3 Ejemplo de funcionamiento.....	58
5.4 Análisis de datos	60
5.5 Logs:	63
6. Presupuesto	64
7. Conclusiones y futuras líneas de investigación y desarrollo.....	65
8. Referencias Bibliográficas	67
9. ANEXO: Código Fuente del Programa	69

Índice de Figuras y Tablas

Figura I.1. Doctor Willem Einthoven (1860 – 1927)	8
Figura I.2. Esquema típico para la adquisición de la señal ECG	9
Figura I.3. Ejemplo de señal ECG disponible en la base de datos de Physionet	10
Figura I.4. Ejemplo de Holter	11
Figura 1.1. Potencial de acción típico de una célula ventricular del miocardio	12
Figura 1.2. Imagen del corazón abierto donde se aprecian los nódulos SA y AV	13
Figura 1.3. Campo eléctrico creado por el dipolo debido a la corriente en una célula del miocardio	15
Figura 1.4. Modelo idealizado para la medición del ECG	16
Figura 1.5. Diferentes estados del vector cardíaco	17
Figura 1.6. Plano de derivaciones frontales	18
Figura 1.7. Las seis derivaciones estándar colocadas en el pecho	18
Figura 1.8. Patrón temporal del vector cardíaco combinado con la geometría de las derivaciones standard	19
Figura 1.9 Características de una señal ECG estándar	20
Figura 1.10. Papel milimetrado con los valores típicos	21
Figura 1.11 Ritmo sinusal normal	22
Figura 1.12. Taquicardia sinusal en las derivaciones I, II y III	23
Figura 1.13. Bradicardia sinusal en la derivación II 50 latidos por minuto	23
Figura 1.14 Arritmia sinusal, nótese las diferencias en el intervalo RR, que marcan la arritmia	24
Figura 1.15. Contracción auricular prematura	25
Figura 1.16. En imagen se indica una contracción ventricular prematura	25
Figura 1.17. Ejemplo de taquicardia ventricular	26
Figura 1.18 ECG de 12 derivaciones mostrando fibrilación auricular a 150 latidos por minuto	26
Figura 1.19. Diferentes clases de bloqueos AV	27
Figura 1.20. Latido de escape ventricular	28
Figura 1.21. Registro 12 derivaciones que muestra un paciente con síntomas isquémicos	29
Figura 1.22. Elevación del segmento ST por isquemia cardíaca	29
Figura 2.1. Señal ECG 10 segundos junto a su periodograma	32
Figura 2.2 PSD (dB/Hz) de 12 derivaciones estándar de una señal ECG de 10 segundos	33

Figura 2.3. Características de una señal ECG estándar	35
Tabla 2.1. Valores normales habituales en ondas ECG	36
Figura 2.4 Señal ECG con <i>baseline wander</i> en azul, en verde el <i>baseline wander</i> extraído con filtrado paso bajo	37
Figura 2.5. Imagen del efecto Gibbs para la transformada de Fourier de la función escalón	40
Figura 3.1. Ejemplo de señal ECG con Baseline wander	41
Figura 3.2. Densidad espectral de potencia del filtro paso alto	42
Figura 3.3. Densidad espectral de potencia filtro paso bajo	43
Figura 3.4. Densidad espectral de potencia filtro notch 50 Hz	43
Figura 3.5. Densidad espectral de potencia notch 60 Hz.....	44
Figura 3.6 Típico esquema de cancelación de ruido mediante filtrado adaptativo	44
Figura 3.7. Esquema básico de un detector QRS.....	45
Figura 3.8. Arriba señal ECG, abajo señal ECG que ha pasado por una etapa de filtrado del preprocesador	45
Figura 3.9. Resultado final del algoritmo gracias a las reglas de decisión.....	46
Figura 3.10. Diagrama de bloques de la etapa de filtrado	46
Figura 3.11. Respuesta en frecuencia del filtro paso bajo.....	47
Figura 3.12. Diagrama de polos y ceros del filtro paso bajo.....	47
Figura 3.13. Respuesta en frecuencia del filtro paso alto	48
Figura 3.14. Diagrama de polos y ceros del filtro paso alto.....	48
Figura 3.15 Respuesta en frecuencia del diferenciador	49
Figura 3.16 Diagrama de polos y ceros del diferenciador	49
Figura 3.17. Algoritmo del detector de pico.....	50
Figura 3.18. Estimación espectral de las diferentes ondas que componen la señal ECG	51
Figura 3.19. Detector QRS con filtrado paso banda y transformada Hilbert	51
Figura 3.20. Ejemplo de transformada de Hilbert sobre señal ecg filtrada paso banda.	52
Figura 3.21. Detector de pico.....	52
Tabla 4.1. Tasas de éxito de los dos algoritmos	53
Tabla 4.2. Tiempos medios de ejecución de los dos algoritmos.....	55
Figura 5.1. Pantalla principal.....	57
Figura 5.2. Señal analizada entre los segundos 12 y 16	59
Figura 5.3. Señal analizada entre los segundos 10 y 50	59
Figura 5.4. Señal analizada entre los segundos 150 y 250	60
Figura 5.5. Herramienta para medir la variabilidad de los latidos	61
Figura 5.6. Herramienta para medir la amplitud de la onda R.....	62

Introducción

Gracias a la ciencia y a la técnica, mucho de los males que sufre la humanidad desde su comienzo han decaído hasta convertirse, al menos en los países del primer mundo, en asuntos anecdóticos. El acceso a los alimentos es más fácil que nunca lo ha sido en la historia, la tasa de mortalidad infantil es prácticamente cero y es raro que personas jóvenes mueran por infecciones en los tiempos que vivimos. Problemas que hace solo doscientos años causaban gran mortalidad en la desarrollada Europa.

Sin embargo, por una especie de equilibrio aristotélico, el progreso ha hecho aparecer algunos males totalmente nuevos y ha dado auge a otros que eran prácticamente una anécdota en los tiempos antiguos. Los tres males más comunes del primer mundo se pueden resumir en lo que se denomina las tres “C”. Coche, Cáncer y Corazón.

Los accidentes de tráfico tienden a afectar al global de la población (aunque de manera más intensa a los jóvenes). El cáncer y las enfermedades cardíacas en cambio suelen aparecer con más frecuencia en personas de mediana o avanzada edad.

Este proyecto se centra en la parte técnica que la medicina moderna usa para tratar las enfermedades del corazón. La ingeniería ha aportado grandes herramientas, más allá de las investigaciones médicas, desde que el doctor holandés Willem Einthoven inventara el primer método práctico para adquirir las señales bilógicas que el corazón produce.



Figura I.1. Doctor Willem Einthoven (1860 – 1927)

Con la herramienta inventada por Einthoven, una nueva rama de la medicina acababa de comenzar. Después de años en los que el comportamiento del corazón era algo de difícil observación, el descubrimiento del Electrocardiograma (ECG, o EKG como lo

nombró originalmente el doctor Einthoven) fue una revolución en el área científica. No sólo el ECG permitía que la señal que genera el corazón fuese visible sino que, además, resultó ser una señal en los casos sanos sorprendentemente predecible. Un médico con entrenamiento podía realizar un diagnóstico sin demasiada dificultad. Localizar una arritmia auricular u otra ventricular, una vez definidos los patrones de tales dolencias en la literatura científica, se convirtió en algo sencillo para la ciencia.

Con el comienzo de los programas de sanidad pública, y el incremento de la esperanza de vida, la incidencia de las enfermedades cardíacas fue en aumento, llegando a convertirse en el problema de salud pública que es hoy en día.

Con la llegada de la computación pronto quedó claro que unas señales relativamente simples de procesar eran un problema ideal para investigar. Inventos tales como los amplificadores, los procesadores digitales de la señal y sistemas de adquisición de datos fueron una gran mejora, ya que permitieron la construcción de aparatos de medida digitales.

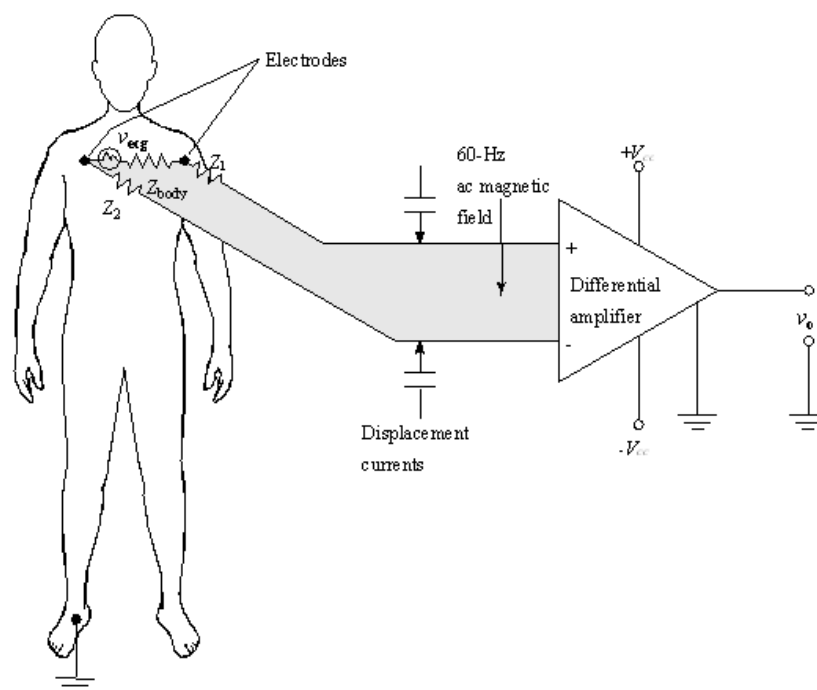


Figura I.2. Esquema típico para la adquisición de la señal ECG

La revolución de la microelectrónica permitió la construcción de mejores aparatos ECG, con una señal adquirida mucho más pura y pantallas para mostrar la señal en vez de los kilométricos rollos de papel que se usaban anteriormente.

A pesar del gran avance logrado, el doctor debía estar pendiente de la monitorización del paciente. Además existen ciertas arritmias que son de complicada localización a simple vista. Incluso dentro de la investigación más básica, el estudio de los registros de señales ECG requería que los investigadores analizaran muchas horas de registros de ECG manualmente para localizar simplemente algunos tramos de interés.

Este problema fue lo que llevó a que en los años ochenta los esfuerzos de universidades americanas, como por ejemplo el MIT (Massachusetts Institute of Technology), se

centraran en la búsqueda de algoritmos que permitieran el análisis automático de las señales ECG.

Durante esta década surgieron algunos de los métodos más usados hoy en día para el análisis de señales ECG. Uno de los algoritmos implementados para este proyecto [1] fue desarrollado en 1985, como una propuesta para la detección automatizada de complejos QRS.

Dichas investigaciones durante la década de los ochenta estuvieron muy limitadas por la velocidad que los microprocesadores de la época alcanzaban. Para realizar estos estudios teóricos y comprobar la viabilidad de los algoritmos, se creó la base de datos de www.physionet.org dependiente del MIT para analizar la viabilidad de los algoritmos.



Figura I.3. Ejemplo de señal ECG disponible en la base de datos de Physionet

No fue hasta los años noventa cuando el incremento de la velocidad en los procesadores digitales de señal permitió implementar de una forma práctica los algoritmos desarrollados una década atrás. Tales algoritmos supusieron la teoría necesaria para dotar de funcionamiento a los aparatos de medición de la actividad cardiovascular conocidos como Holters.



Figura I.4. Ejemplo de Holter

Los Holters permiten la grabación de registros cardiacos de pacientes mientras estos realizan sus tareas cotidianas. Las señales que se obtienen de los pacientes representan una fuente de gran valor para diagnósticos de problemas tales como soplos cardiacos y ciertas arritmias que se producen durante muy corto tiempo y de manera muy aleatoria.

Debido a las limitaciones de alimentación que pueden presentar al funcionar durante largos periodos. Es necesario que los algoritmos sean lo más óptimo posible para lograr maximizar la duración de la batería.

Actualmente, la potencia de la computación ha llevado a que los estudios sobre procesamiento de señales biológicas se centren sobre campos como las técnicas de **Machine Learning**, análisis mediante fractales, algoritmos de aprendizaje supervisados y no supervisados y algoritmos propios de **Data Mining**.

¿En qué se centra este proyecto?

Este proyecto pone el foco en desarrollar una herramienta de fácil uso en Matlab que permita el análisis del rendimiento de diferentes algoritmos. Para ello provee una interfaz gráfica de fácil uso para analizar las señales y un soporte para que el usuario pueda desarrollar sus propios algoritmos.

En este proyecto se implementa un algoritmo considerado clásico del procesamiento de señales de ECG y en segundo lugar se desarrolla desde cero un algoritmo que resulta más adecuado para el procesamiento de señales en dispositivos tipo Holter ya que presenta una complejidad computacional menor.

1. Principios Básicos del Electrocardiograma

1.1. El proceso celular subyacente en el ECG.

Cada latido del corazón es disparado por un potencial de acción que se origina de forma rítmica en el interior del corazón y es conducido rápidamente a través de dicho órgano, lo cual produce una contracción coordinada. Como otros tejidos eléctricamente activos, como por ejemplo nervios y músculos esqueléticos, las células del miocardio tienen un potencial transmembrana típico de alrededor de -80 a -90 mV con respecto al líquido extracelular.

La membrana celular controla la permeabilidad al número de iones, incluyendo sodio, potasio, calcio y cloruro. Estos iones pasan a través de la membrana mediante unos canales de ion específicos que pueden estar abiertos o cerrados. Las variaciones de la conductividad de la membrana debido al cierre y apertura de estos canales generan cambios en el potencial de acción a lo largo del tiempo (Figura 1.1). La forma que adopta el potencial acción a lo largo del tiempo se representa en la siguiente figura.

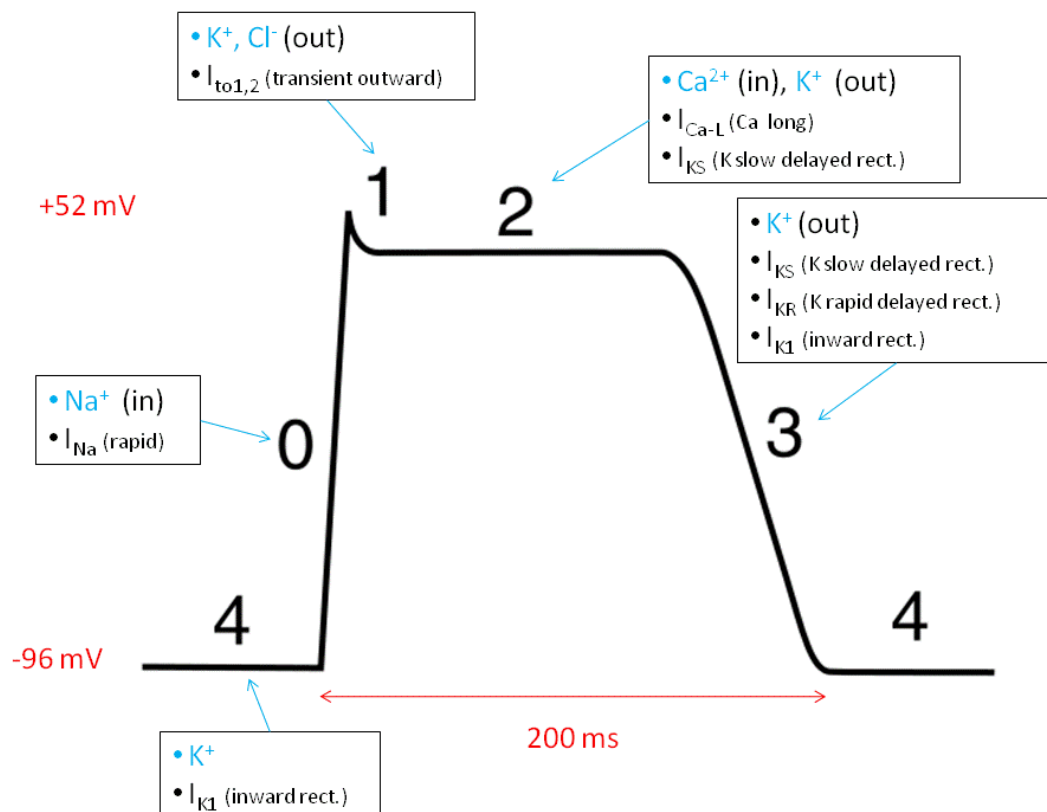


Figura 1.1. Potencial de acción típico de una célula ventricular del miocardio

Existen 4 fases diferenciadas en un potencial de acción típico:

- Fase 1: Representa una repolarización inicial que es causada por la apertura de un canal de potasio.
- Fase 2: Durante la fase 2 hay un balance entre las corrientes entrantes de iones de Calcio y las corrientes salientes de Potasio, causando una zona prácticamente plana en el potencial de acción y un retraso en la polarización.
- Fase 3: Es una fase compleja y en la cual gran parte de mecanismos están implicados. Corresponde a la repolarización. La conductividad del Potasio se incrementa, lo que lleva a la repolarización de la célula. Además de esto, hay un decrecimiento de la conductividad del calcio lo cual también contribuye a la repolarización celular.
- Fase 4: También llamada la fase de descanso se caracteriza por la apertura de los canales de potasio y el potencial negativo de la membrana.

Antes de que el potencial de acción sea propagado, es necesario que sea iniciado por un **“marcapasos”**. Lo que significa que debe existir algún tipo de células que tengan la propiedad de generar automáticamente despolarizaciones lo cual marcaría el ritmo al resto de células del corazón.

Dichas células se encuentran en el nódulo sinoauricular (Nódulo SA), en el nódulo auriventricular (Nódulo AV) y en ciertas partes entre las aurículas y los ventrículos.

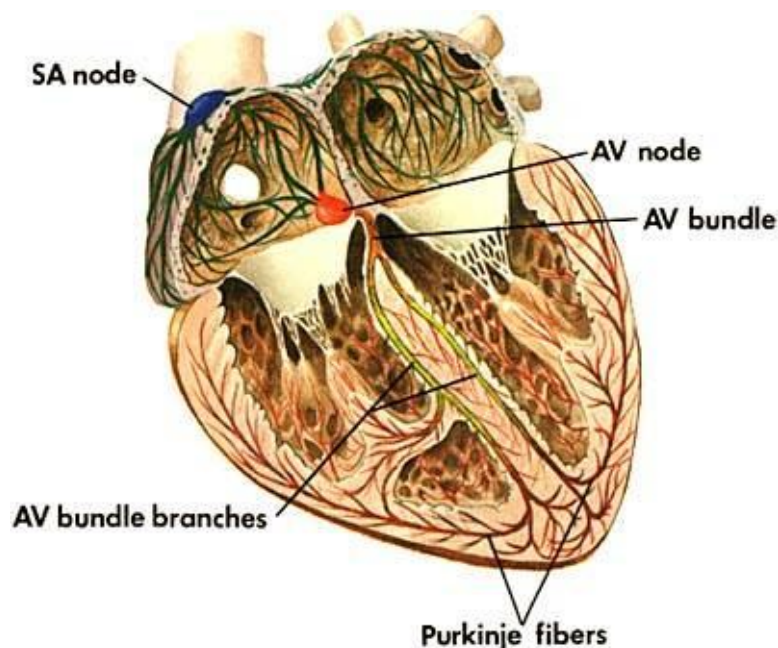


Figura 1.2. Imagen del corazón abierto donde se aprecian los nódulos SA y AV

En esta clase de células la fase 4 no es estable, sino que muestra una despolarización espontánea, incrementándose su potencia transmembrana lentamente hacia cero debido a un pequeño flujo de iones de sodio y calcio que entran en estas células a través

de los canales correspondientes. Cuando el potencial de las células alcanza un nivel umbral, la célula desarrolla un potencial de acción similar a la fase 0 anteriormente descrita, pero que se lleva a cabo a través de un intercambio de iones de calcio a mucho menor ritmo. Después del potencial de acción, el potencial de la membrana vuelve al momento de descanso o fase 4 y el ciclo está listo para repetirse una y otra vez.

La frecuencia de estas células no es la misma para todas ellas, usualmente:

- Nódulo SA; De 60 a 100 latidos por minuto.
- Nódulo AV; De 40 a 50 latidos por minuto.
- Músculos ventriculares; Alrededor de 20 a 40 latidos minuto.

Bajo condiciones normales de funcionamiento, el nódulo SA determina el latido del corazón. Sin embargo en algunas patologías la tasa de latidos de los nódulos AV o los músculos ventriculares puede exceder la de los del nódulo SA y entonces los otros marcadores son los que determinan el ritmo de latido del corazón.

Un potencial de acción, una vez iniciado en una célula cardíaca, se propagará a lo largo de la membrana celular hasta que la célula este totalmente despolarizada. Las células de miocardio tienen la propiedad única de transmitir potenciales de acción de una célula a otra adyacente por medio de una corriente de difusión (sin sinapsis electroquímicas). Las células están realmente separadas, sin conexión física entre ellas, pero lo suficientemente cerca como para transmitir entre ellas corriente y tensión. Las corrientes iónicas fluyen de célula a célula, lo que conlleva que el impulso generado se propague por el corazón, resultando en una contracción mecánica coordinada.

Por ejemplo, los marcapasos artificiales introducen impulsos de despolarización con un electrodo colocado en el interior del ventrículo derecho, dando lugar a contracciones mecánicas del corazón idénticas a las producidas naturalmente.

Hay que aclarar que esta explicación es una explicación bastante resumida. Para mayor profundidad consultar textos más detallados [2] [3]

1.2. Física Básica de la electrocardiografía

Como resultado de la actividad eléctrica de las células, se establecen diferencias de potencial en la superficie de la piel que pueden ser medidas mediante el uso de equipamiento adecuado.

La grabación de estos potenciales de superficie produce la señal que comúnmente conocemos como electrocardiograma. El modelo matemático más simple para relacionar el generador de señal cardíaca con los potenciales medidos en la superficie de la piel es el conocido como modelo del dipolo simple. El modelo es realmente sencillo y proporciona un entorno para el estudio de la electrocardiografía y la vectorcardiografía. Aun teniendo en cuenta que hay desarrollados modelos mucho más complejos [3] [4].

El modelo del dipolo tiene dos componentes, una representación de la actividad eléctrica del corazón (el dipolo en sí mismo), y la geometría y las propiedades eléctricas del cuerpo. Existe una corriente intracelular generada en la dirección de propagación.

Esta es la fuente elemental de corriente de la superficie de ECG, referida como el dipolo de corriente. Existe también una corriente de la misma magnitud extracelular fluyendo en dirección contraria a la dirección de propagación, lo cual permite la conservación la carga.

Todos los flujos de corriente tienden a cerrarse, formando un dipolo eléctrico que genera un campo eléctrico. La actividad total del corazón en cualquier instante puede ser representada por una distribución de dipolos.

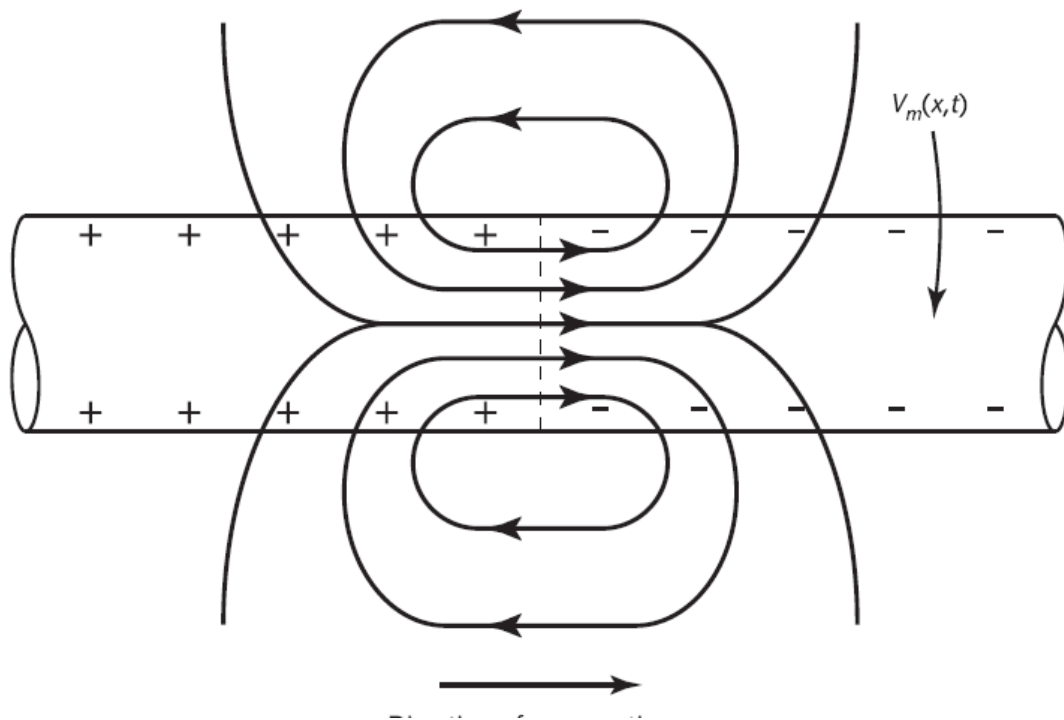


Figura 1.3. Campo eléctrico creado por el dipolo debido a la corriente en una célula del miocardio

Si el corazón estuviera suspendido en medio conductor homogéneo e isotrópico y fuera observado desde una distancia lo suficientemente larga comparado con su tamaño, en ese caso podríamos suponer que todas las corrientes se encuentran en el mismo punto en el espacio y la actividad eléctrica total puede ser representada como un dipolo. Normalmente este dipolo es denominado como $\mathbf{M}(t)$ y se considera dependiente del tiempo. En cada ola de despolarización que se expande a través del corazón el vector $\mathbf{M}(t)$ cambia de magnitud y de dirección.

Las distribuciones superficiales de corriente y potenciales dependen de las propiedades eléctricas del torso. Como primera aproximación, se supone al torso como lineal homogéneo e isotrópico. Tratándolo como un medio de dichas características, y aproximando su forma a la de una esfera de radio R y conductividad σ , la fuente se puede representar como un dipolo de corriente, localizado en el centro de la esfera, que varía lentamente con respecto al tiempo. El campo eléctrico estático, la densidad de corriente y el potencial eléctrico en todos los lugares dentro de la esfera y su superficie están relacionados dinámicamente con el vector $\mathbf{M}(t)$ en cualquier momento. Los

términos reactivos de la impedancia de los tejidos pueden ser ignorados. Resolviendo la ecuación de Laplace obtenemos la distribución de potencial en el torso.

$$\Phi(t) = \cos\theta(t)3|M(t)|/4\pi\sigma R^2$$

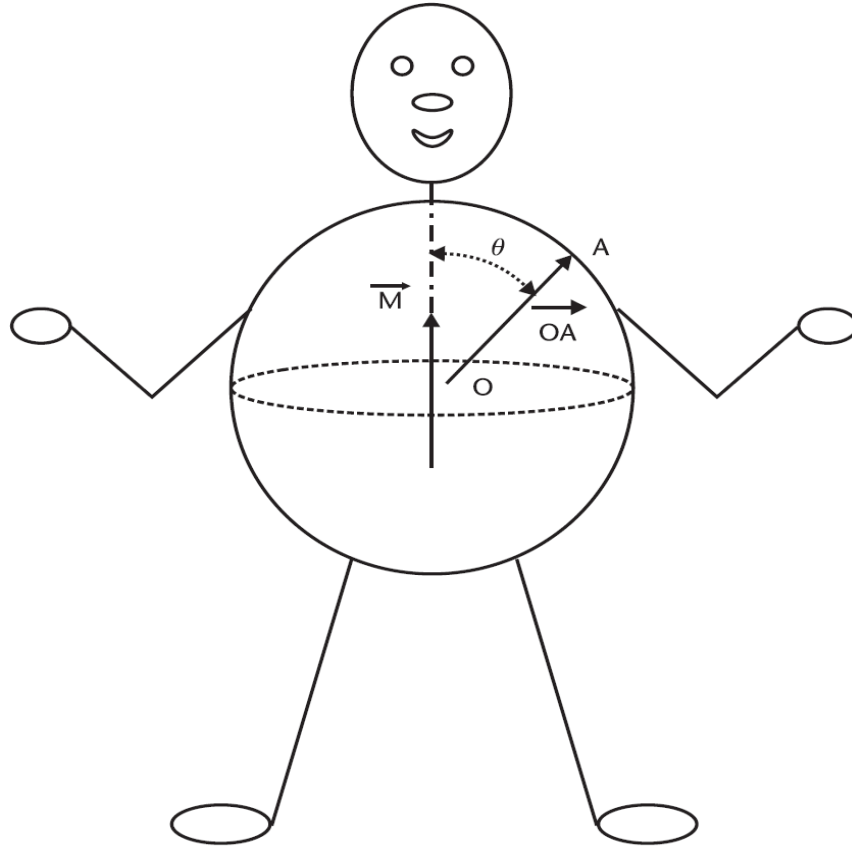


Figura 1.4. Modelo idealizado para la medición del ECG

Donde $\theta(t)$ es el ángulo entre la dirección del vector $M(t)$ y el vector OA de la derivación tal como se muestra en la figura. OA es el vector origen al punto de observación. Módulo de M es la magnitud del vector corazón.

La diferencia de potencial entre puntos de la superficie del torso sería:

$$V_{AB}(t) = M(t) \cdot L_{AB}(t)$$

Donde $L_{AB}(t)$ es el vector que une los puntos de la derivación.

Es útil para las mediciones reales definir un terminal de referencia central mediante la ponderación de tres derivaciones en los miembros:

$$\Phi_{CT}(t) = \Phi_{RA}(t) + \Phi_{LA}(t) + \Phi_{LL}(t)$$

Donde RA indica brazo derecho, LA indica brazo izquierdo y LL indica pierna izquierda. Los cuales son generalmente puntos típicos donde se hacen mediciones de electrocardiograma

1.3 La señal de electrocardiograma

Durante las diferentes fases del funcionamiento del corazón. El vector $M(t)$ o vector cardíaco varía en función del momento del ciclo cardíaco en el que nos encontremos. La siguiente figura ilustra las diferentes trayectorias de dicho vector en un corazón sano:

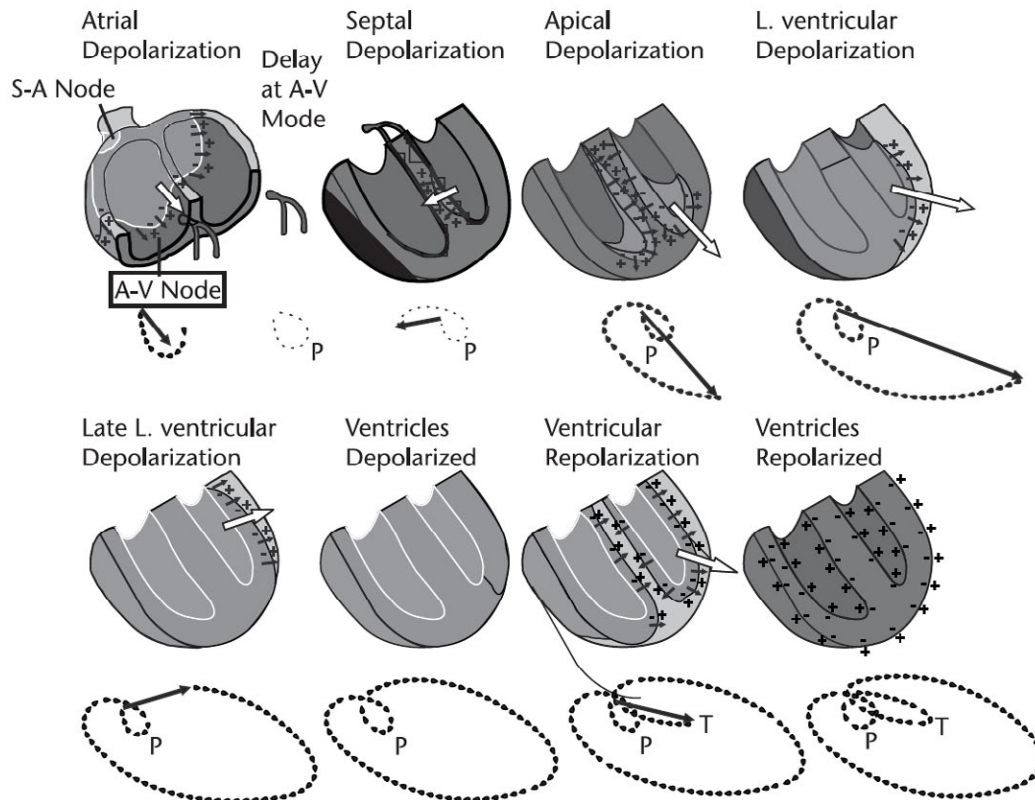


Figura 1.5. Diferentes estados del vector cardíaco

Como se puede apreciar se muestran nueve fases. La figura muestra donde se origina el latido (nódulo SA), un retraso en el nódulo AV y una conducción acelerada de despolarización. La línea punteada debajo de cada ilustración representa la trayectoria precedente de los vectores cardíacos.

Para mostrar los vectores en la típica forma de la señal ECG que es comúnmente conocida, el vector cardíaco, el cual se expande, contrae y rota en un espacio tridimensional es proyectado en 12 líneas. Cada derivación, la cual se coloca en estas orientaciones, obtiene la magnitud del vector cardíaco en la dirección en la que se encuentra cada derivación en cada instante de tiempo. Las seis derivaciones precordiales informan de la actividad en el plano horizontal. En la práctica esto requiere de seis electrodos que son colocados alrededor del torso. La señal ECG representa la diferencia entre cada uno de esos electrodos y el terminal central.

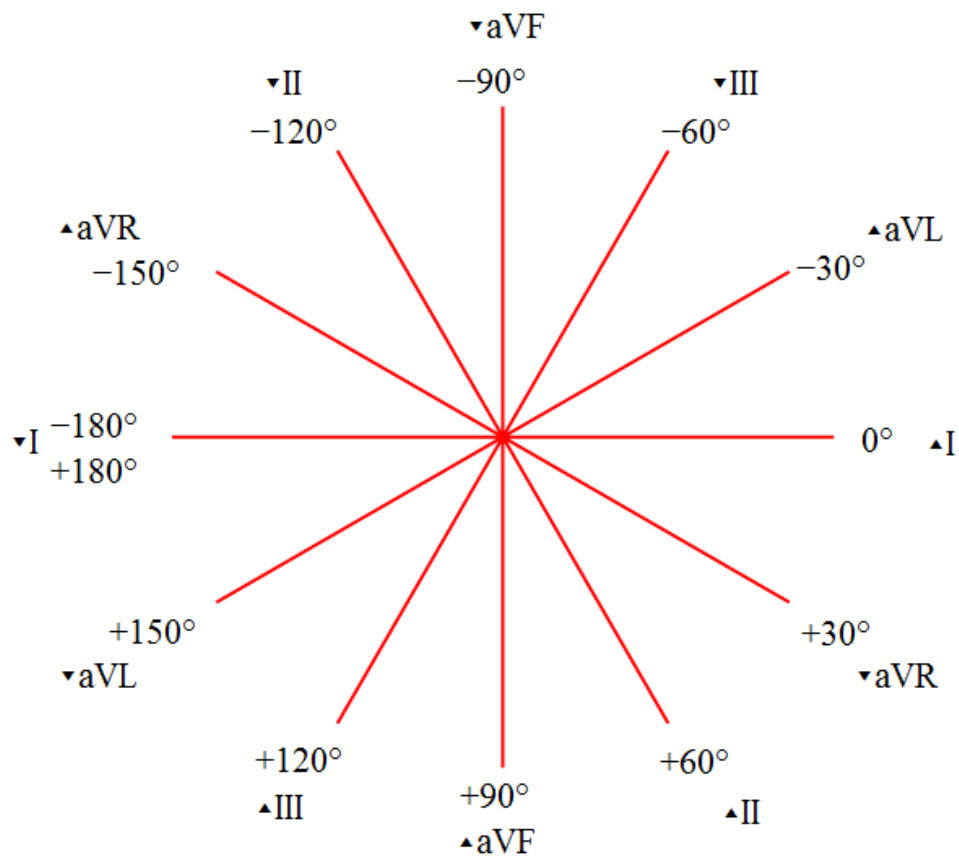


Figura 1.6. Plano de derivaciones frontales

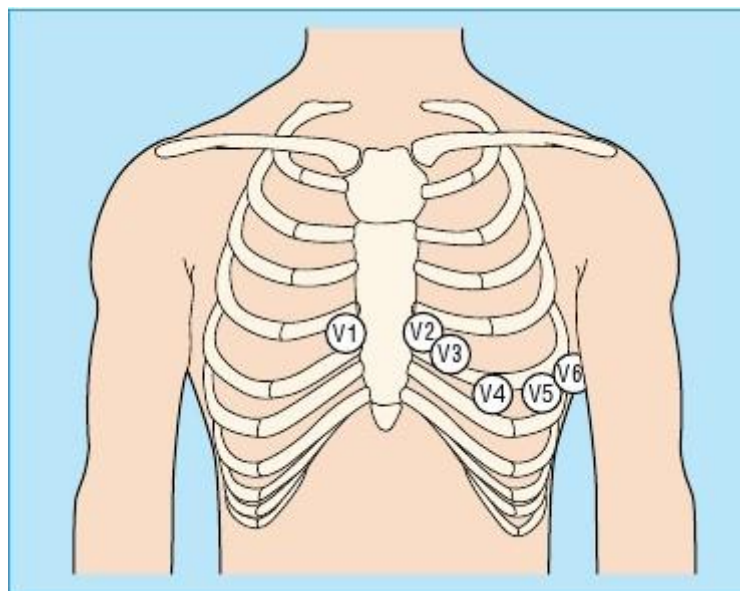


Figura 1.7. Las seis derivaciones estándar colocadas en el pecho

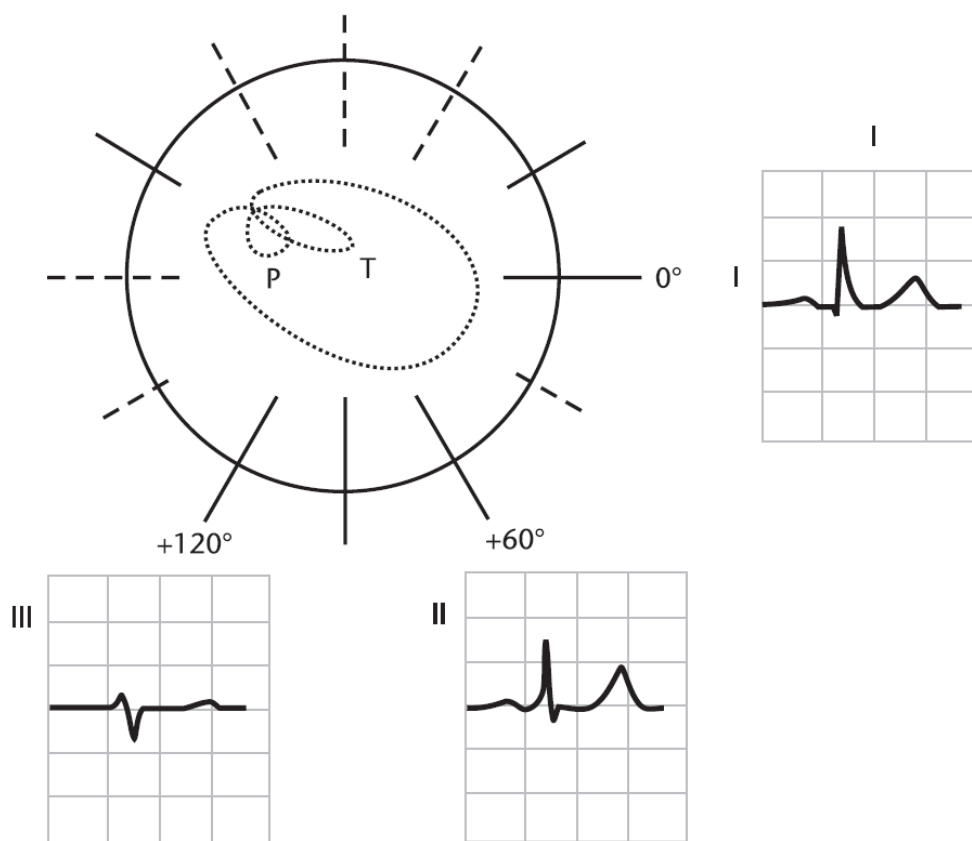


Figura 1.8. Patrón temporal del vector cardíaco combinado con la geometría de las derivaciones estándar

Además de estas derivaciones, se colocan electrodos en los cuatro miembros del sujeto y un terminal central tal como se vio anteriormente.

Las diferencias de potencial entre los electrodos en los miembros y el terminal central son para las otras tres derivaciones del ECG, como se muestra en la figura 1.6:

- Derivación I: Diferencia entre brazo izquierdo y brazo derecho.
- Derivación II: Diferencia entre el pierna izquierda y el brazo derecho.
- Derivación III: Diferencia entre pierna izquierda y brazo izquierdo.

Nótese que las derivaciones aumentadas (denominadas con “a”) representan el potencial de un miembro dado con respecto a la media de potenciales de los otros dos miembros. aVF es la diferencia entre la pierna izquierda y una media de las derivaciones de los brazos; aVR es la diferencia entre el brazo derecho y la media de la pierna izquierda y el brazo izquierdo, y aVL es la diferencia entre el brazo izquierdo y la media del brazo derecho y la pierna izquierda. Dado que hay doce derivaciones que representan una imagen tridimensional de la actividad cardíaca, existe bastante información redundante. Sin embargo, este exceso de información repercute en una representación más sencilla de entender para las personas y compensa deficiencias en la colocación de los electrodos (obviamente el cuerpo humano real está muy lejos de parecerse al torso esférico modelado tal como se ve en la figura 1.4).

La posición instantánea del vector cardíaco se combina con la geometría de las derivaciones del plano frontal, tal como se muestra en la figura 1.8. El resultado de las

derivaciones I, II y III resulta, en un corazón sano y una situación del individuo normal, en la señal típica de ECG comúnmente conocida.

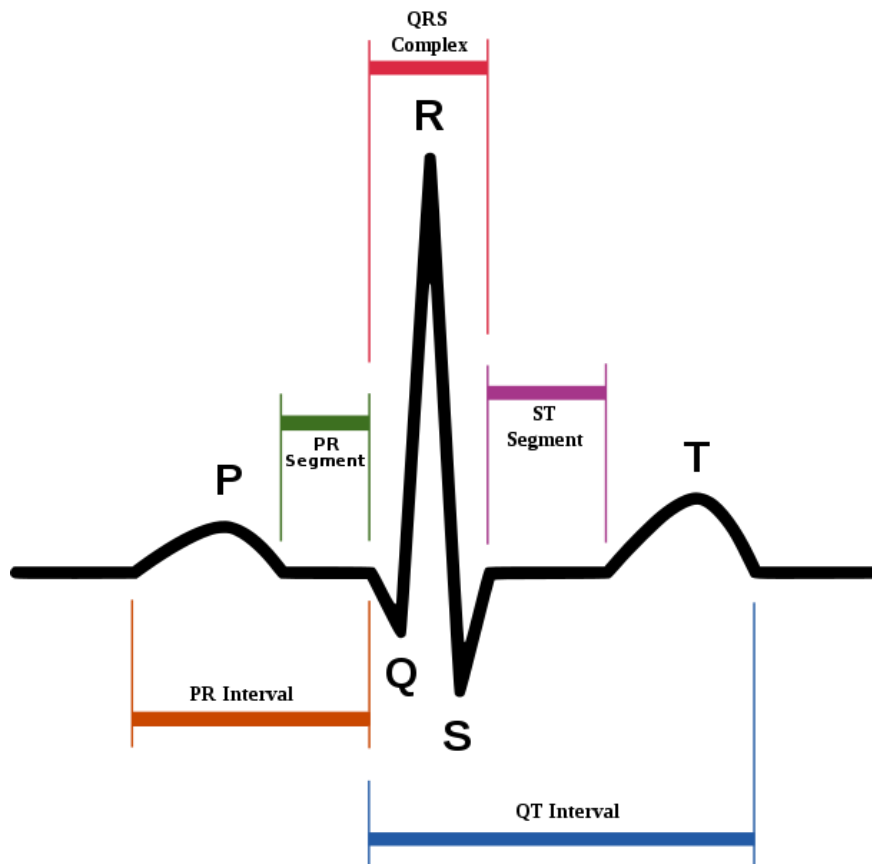


Figura 1.9 Características de una señal ECG estándar

Nótese que el eje del complejo QRS es perpendicular al de la derivación isoelectrica (la derivación con fuerzas iguales en dirección positiva y negativa). Cambios significativos en el eje del complejo QRS puede ser un indicativo de problemas cardiacos.

La figura 1.9 muestra las características normales de un electrocardiograma clínico, entre las que se encuentran los diferentes tipos de ondas, tiempo que transcurre entre ellas y duración de las mismas. Cada una de estas ondas características recibe un nombre por convención con las letras P, Q, R, S y T (a veces también existe una onda U posterior a la T pero es de difícil identificación).

Los intervalos también se muestran en la imagen 1.9, y corresponden al tiempo entre las diferentes ondas. También existe aunque no está representado en la imagen el intervalo RR, que es el tiempo transcurrido entre dos ondas RR, lo que nos ayudará a obtener el número de latidos por minuto.

Para el estudio del ECG clínico se usa un papel milimetrado especial que permite la visualización rápida de las medidas de tiempo y magnitud de las diferentes ondas.

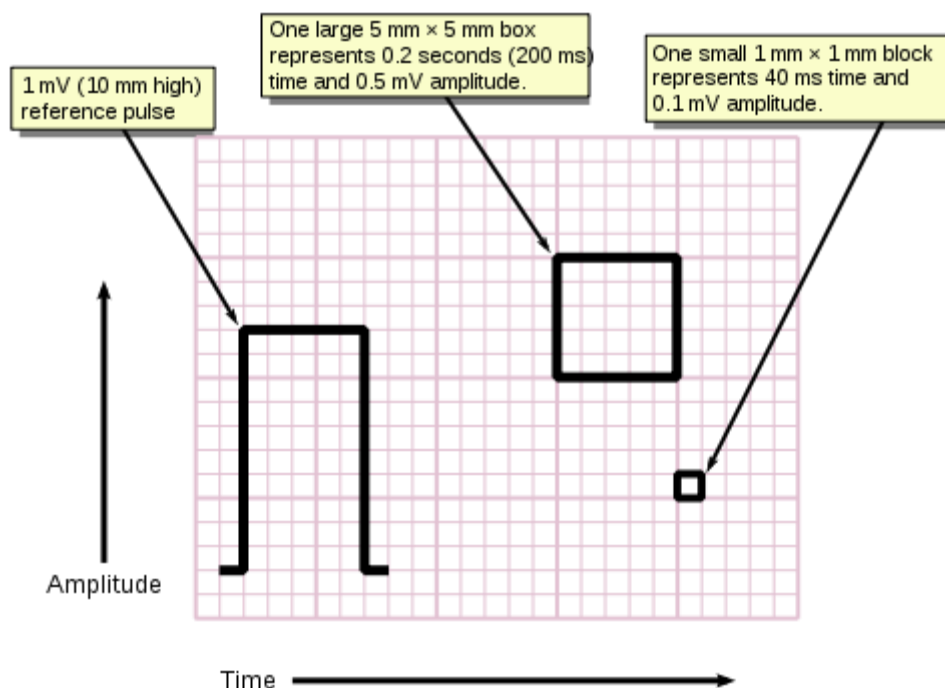


Figura 1.10. Papel milimetrado con los valores típicos

1.4 Patrones anormales en la señal ECG

Los clínicos que usan el electrocardiograma como una herramienta de diagnóstico buscan determinar funcionamientos anormales en el corazón en base a la medición de diferencias de potencial en la superficie de la piel. Aun así el ECG de superficie tiene limitaciones como herramienta de medición. Dada una distribución de potenciales de superficie no podemos determinar precisamente el comportamiento electrofisiológico del corazón debido a que el problema no tiene una única solución como demostró Hermann Von Helmholtz en 1853. Generalmente no es posible determinar las características de un generador de corriente con solo medidas del potencial en la superficie. Debido a estas limitaciones inherentes para una completa caracterización de la actividad cardíaca se necesita un estudio invasivo.

A pesar de las limitaciones, el ECG de superficie es realmente útil para diagnosticar patologías que se vean reflejadas en el comportamiento eléctrico del corazón y solamente en casos muy específicos se recomienda el uso de técnicas invasivas.

A continuación se describirán las características del ritmo normal (llamado ritmo sinusal) y de diferentes anomalías en el ritmo, así como una clasificación de distintos tipos de arritmias cardíacas.

- Ritmo sinusal normal: Es el ritmo de la señal ECG de un paciente sano. Muestra la forma que se representa en la figura 1.9, donde los intervalos son constantes. Los latidos por minuto se encuentran entre los 60 y los 100, la onda P debe estar presente y el ancho del complejo QRS debe ser menor de 0.12s.

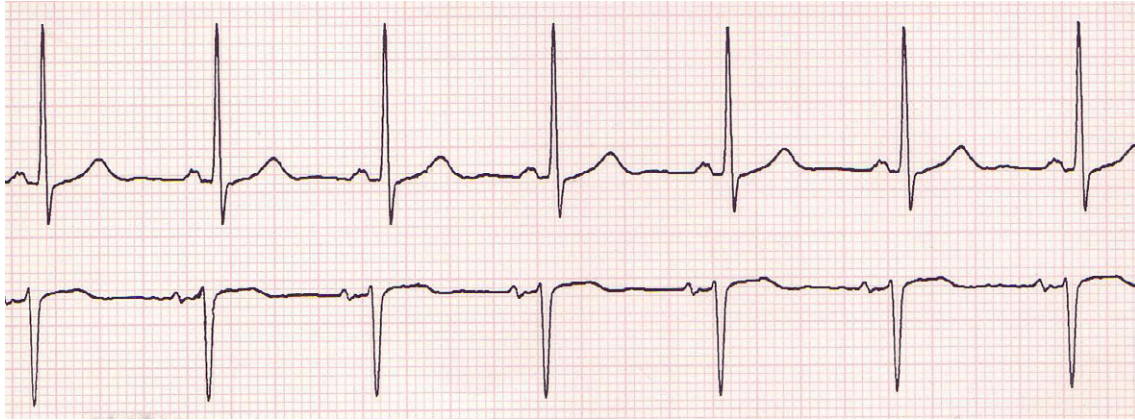


Figura 1.11 Ritmo sinusal normal

- Taquicardia: trastorno cardíaco caracterizado por un aumento en la frecuencia de los latidos del corazón. Se origina en el nódulo sinusal y comienza y termina generalmente de forma gradual. Entre las posibles causas de su aparición se encuentran:
 - o Estrés
 - o Fiebre
 - o Ansiedad
 - o Ejercicio
 - o Hipotensión arterial
 - o Insuficiencia cardíaca congestiva
 - o Deshidratación
 - o Hipertermia maligna
 - o Anemia
 - o Hipertiroidismo
 - o Feocromocitoma
 - o Sepsis
 - o Embolismo pulmonar
 - o Cardiopatía isquémica aguda o infarto agudo de miocardio
 - o Enfermedad pulmonar crónica
 - o Ingesta de estimulantes como la cafeína, nicotina o cocaína
 - o Circulación hiperdinámica

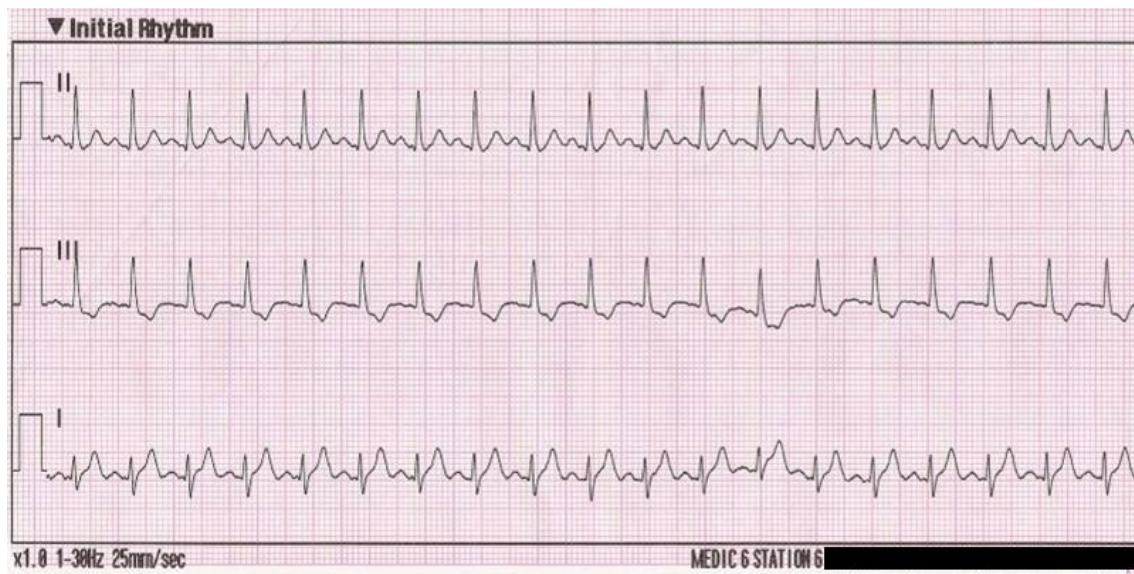


Figura 1.12. Taquicardia sinusal en las derivaciones I, II y III

- Bradicardia: Es el nombre por el cual se conoce a la situación del corazón en que los latidos descienden por debajo de los 60 latidos por minuto, aun teniendo en cuenta que solamente por debajo de los 50 latidos por minuto pueden aparecer problemas. La bradicardia no es necesariamente problemática y aparece en deportistas y adultos jóvenes. Algunas de las causas de la bradicardia son las siguientes:
 - o Hipotermia.
 - o Hipotiroidismo.
 - o Efecto de algunas drogas, como los beta-bloqueantes.
 - o Efecto secundario de infecciones tales como la difteria, fiebre reumática y la miocarditis vírica.

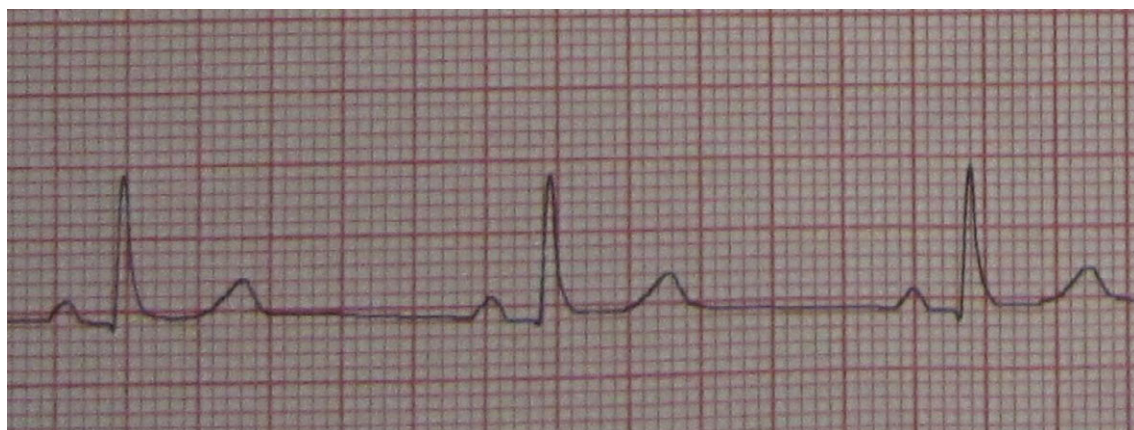


Figura 1.13. Bradicardia sinusal en la derivación II 50 latidos por minuto

1.5 Arritmias Cardíacas

Una arritmia puede ser definida como cualquier ritmo cardíaco anormal. Uno de los tipos de arritmia ocurre cuando el disparo de despolarización se origina fuera del nódulo SA, en otra parte del miocardio (lo que se conoce como despolarización ectópica).

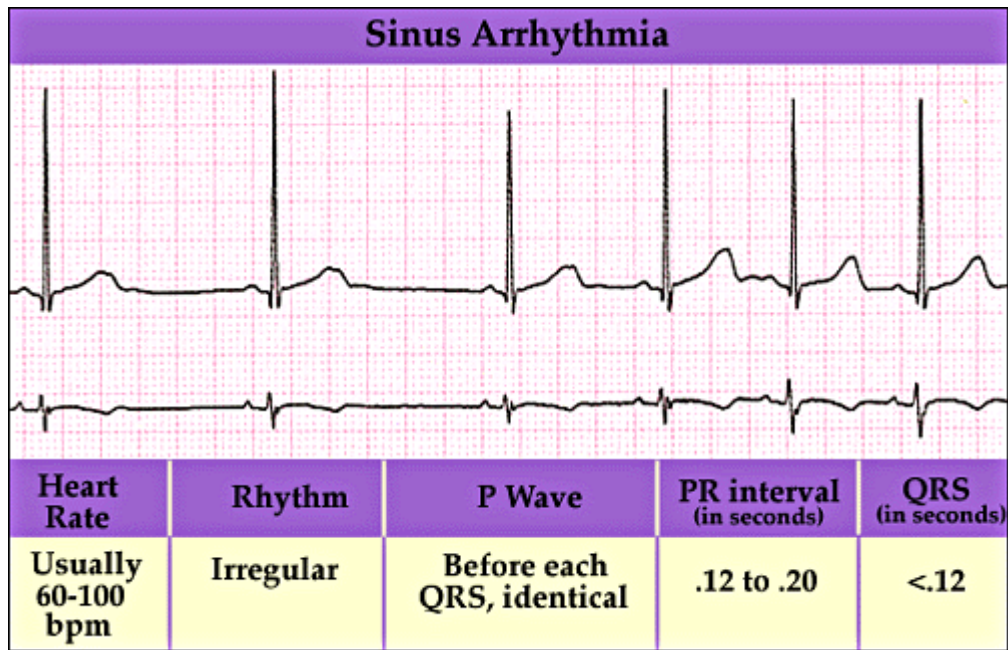


Figura 1.14 Arritmia sinusal, nótese las diferencias en el intervalo RR, que marcan la arritmia

Las causas comunes de la ectopia incluyen el efecto de determinadas drogas (por ejemplo cafeína), una infección viral del miocardio o una inflamación o daño de una parte del corazón (por ejemplo isquemia). Cuando el latido ectópico se genera en las aurículas conduce a un latido auricular prematuro, también conocido como contracción auricular prematura. Cuando se origina en los ventrículos, conduce a la contracción ventricular prematura.

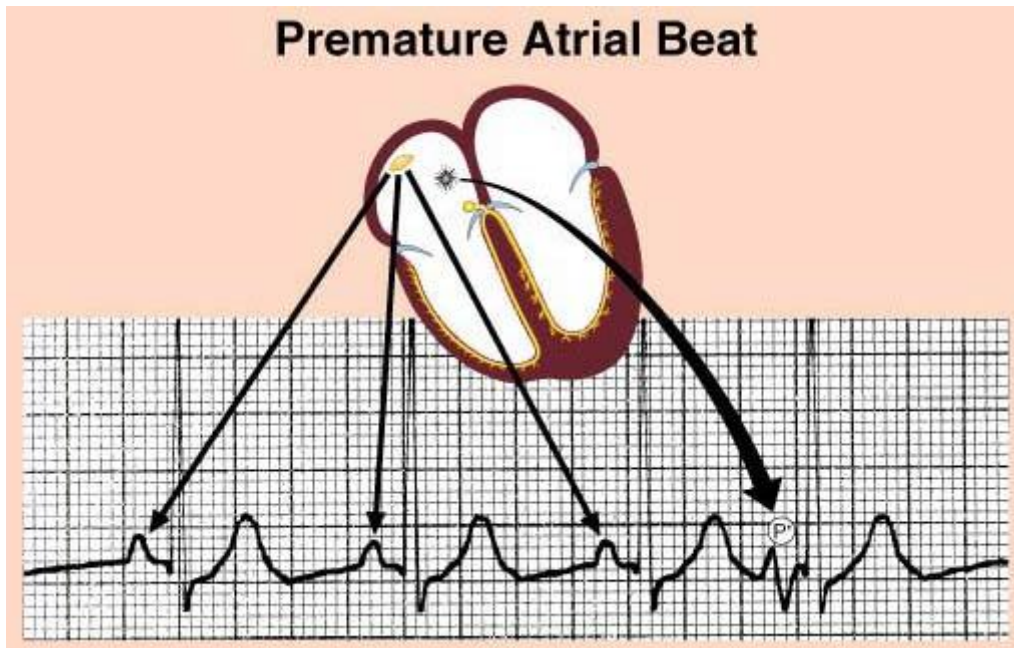


Figura 1.15. Contracción auricular prematura

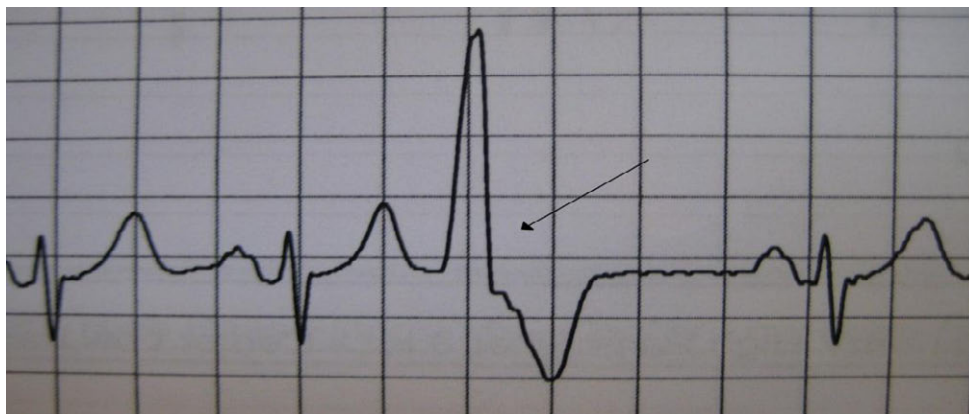


Figura 1.16. En imagen se indica una contracción ventricular prematura

Se puede observar en la figura 1.16 que los latidos ventriculares ectópicos son muy diferentes de los latidos normales. La difusión del frente de onda de uno de estas patologías puede ser hacia atrás, como cuando el potencial de acción empieza en el vértice del corazón en vez de en el septum.

El frente de onda de despolarización puede moverse en muy diferentes direcciones en comparación con el típico vector cardíaco. Además, el latido ectópico es típicamente más amplio debido a que su frente de onda se propaga lentamente a través del miocardio en vez de rápidamente a través del sistema de Purkinji.

Después de que un frente de onda se ha propagado a todas las partes del corazón, el miocardio queda temporalmente despolarizado. Después de una pausa, los tejidos se repolarizan y un mecanismo sinusal normal puede precipitar el siguiente latido. Las condiciones que provocan un latido ectópico pueden persistir, pero el latido ectópico en sí mismo queda atrás en el tiempo, aunque a veces un patrón de latidos ectópicos puede desarrollarse. Por ejemplo patrones regulares de contracciones ventriculares prematuras.

Más problemático es el caso en el que el frente de onda se propaga de una forma casi estable, circulando repetidamente a través del corazón conduciendo a repetidas olas de despolarización en los tejidos a un ritmo anormalmente rápido. Cuando esto ocurre en el corazón recibe el nombre de arritmia reentrante. Un ejemplo clásico de este tipo de arritmias es la llamada taquicardia ventricular, mostrada en la figura de abajo.

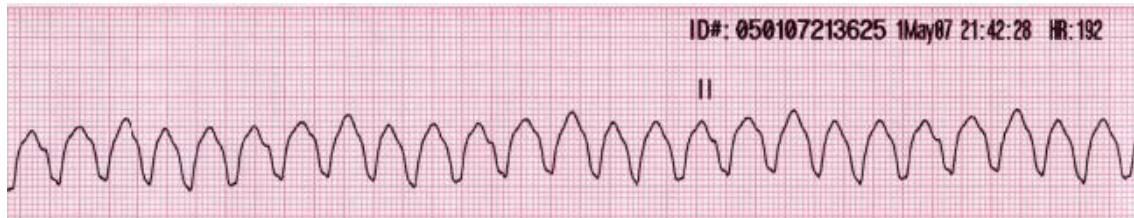


Figura 1.17. Ejemplo de taquicardia ventricular

La taquicardia ventricular puede ser fatal en muy poco tiempo ya que la tasa de despolarización puede hacer imposible que el corazón pueda bombear. En algunos casos la taquicardia ventricular puede acabar con la vida en segundos o minutos pero en el otro extremo puede permanecer en un nivel compatible con la vida.

El criterio para identificar alguna anomalía como una taquicardia ventricular son tres o más latidos ventriculares ectópicos a una tasa de latidos por minuto mayor de 100. Si la taquicardia ventricular termina en 15 segundos (o 30 según otras convenciones) se conoce como taquicardia ventricular no sostenida; de otra forma se considera como taquicardia ventricular sostenida. Debido a las graves consecuencias médicas de la taquicardia ventricular han sido investigadas experimentalmente, clínicamente y teóricamente.

En algunos casos, el frente de onda de despolarización puede descomponerse en incontables frentes más pequeños que circulan aleatoriamente sobre el miocardio. Esto lleva a una rotura total de la contracción coordinada y el miocardio comenzará a temblar. A este suceso se le llama fibrilación.

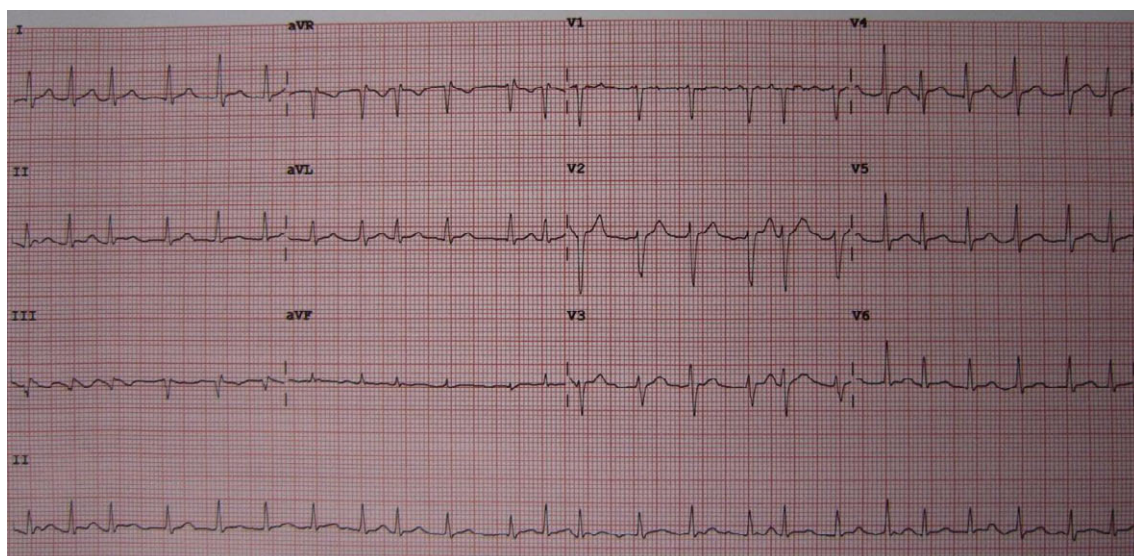


Figura 1.18 ECG de 12 derivaciones mostrando fibrilación auricular a 150 latidos minuto

La otra categoría de arritmias está relacionada con un ritmo de los latidos excesivamente lento y bloqueos anormales del frente de propagación de la onda. Por ejemplo, un envejecido nódulo SA puede hacer que el ritmo de latidos sea demasiado lento, llevando a una baja presión sanguínea y debilidad o desmayos en el paciente.

Asumiendo que esto no es producido por una excesiva medicación, esos síntomas pueden requerir la implantación de un marcapasos artificial. El nódulo AV puede en ocasiones también desarrollar patologías de conducción que enlentecen el ritmo ventricular: Puede llevar a fallo en la conducción de algunos frentes de onda auriculares (bloqueo AV de segundo grado) o puede llevar al fallo de todas las ondas auriculares (lo que se denomina bloqueo AV de tercer grado). Si el paciente no ha sido sobre medicado, la mayoría de los bloqueos AV de tercer grado y algunos de los de segundo grado requieren de terapia de marcapasos. En los bloqueos AV de primer grado, el nódulo AV conduce el frente de onda auricular con un retraso anormal. Los bloqueos AV de primer grado no enlentecen los latidos medios ventriculares.

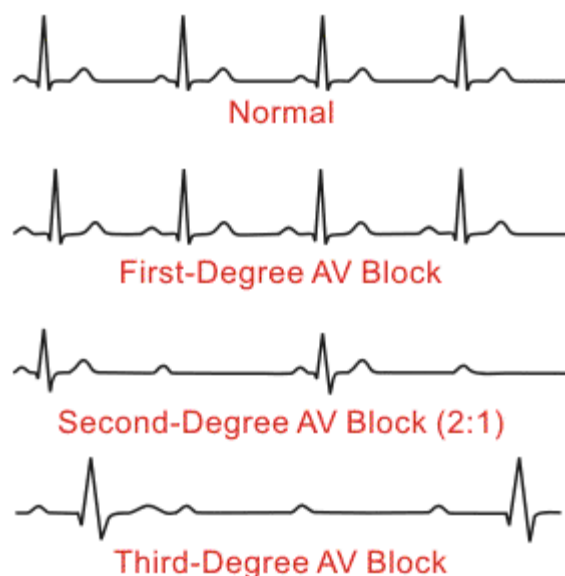


Figura 1.19. Diferentes clases de bloqueos AV

También pueden fallar algunas fibras de conducción de los ventrículos, tal que las ondas de despolarización tienen que alcanzar algunas zonas del ventrículo izquierdo a través de una propagación “musculo – musculo” que es mucho más lenta. Hay un conjunto de cambios asociados a diferentes fallos en los conductos de conducción.

Estos bloqueos tienen normalmente un efecto mínimo en la eficacia de bombeo. Sin embargo pueden cambiar dramáticamente la trayectoria del vector cardiaco y por consiguiente de la superficie ECG. Pueden también enmascarar otros cambios indicativos de patología en el ECG (por ejemplo la isquemia). En algunos casos estas anormalidades en la conducción indican alguna otra patología subyacente de gran importancia (por ejemplo un embolismo pulmonar).

El asunto de las bradiarritmias y los bloqueos cardíacos lleva al asunto de los latidos de escape (ver la figura 1.20)

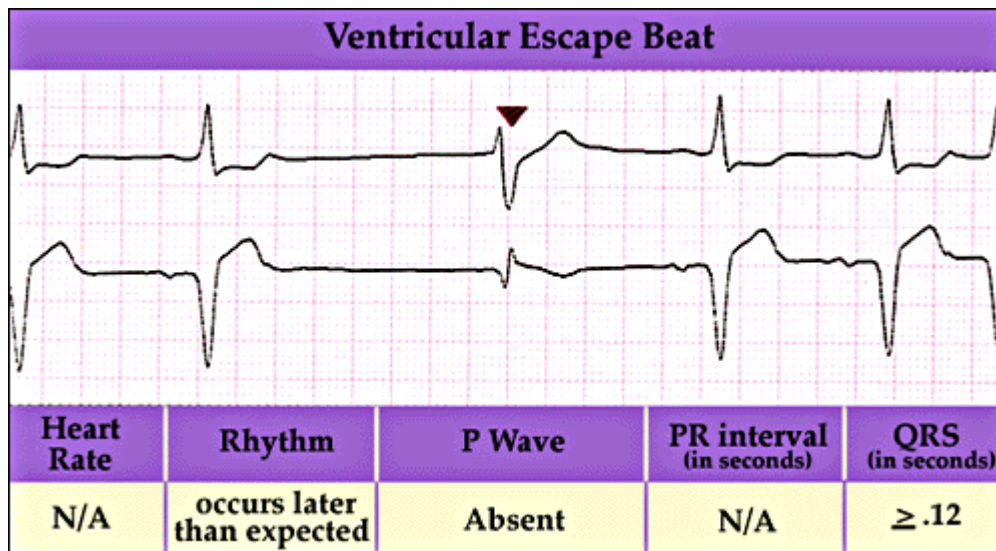


Figura 1.20. Latido de escape ventricular

Un latido de escape es similar a un latido ectópico, en el que la inicialización de la despolarización sucede fuera del nódulo SA. Sin embargo la diferencia es que un latido de escape implica fallos en el sistema eléctrico del corazón. Hay una red de células cardíacas listas para iniciar los latidos del corazón, tanto que las funciones vitales (por ejemplo marcar el ritmo del corazón) no están relegadas a una microscópica colección de células en el nódulo SA. Las células de este sistema tienen intrínsecos ritmos del disparador que son más lentos que los del nódulo SA. Entonces mientras el nódulo SA tiene la oportunidad de marcar el ritmo del corazón, las otras regiones iniciaran los latidos del corazón si ha habido una pausa demasiado prolongada en la señal del nódulo SA. Por tanto, un latido ventricular que ocurra en la configuración de un bloqueo del corazón de tercer grado se nombra como latido de escape ventricular, y representa una respuesta apropiada a la falta de otro disparador que controle el ritmo cardíaco.

Cuando latidos no sinusales aparecen en el ECG, es importante diferenciar latidos ectópicos (donde la patología tiene un ritmo aberrante) de los latidos de escape (donde la patología tiene una conducción anormal), porque los tratamientos en ambos casos son sustancialmente diferentes.

1.6 Otras anomalías: isquemia

La señal ECG puede revelar anomalías metabólicas en el miocardio. La más significativa de dichas anomalías es la isquemia, que se produce cuando parte del miocardio no está recibiendo suficiente flujo sanguíneo, frecuentemente causando por enfermedades en las arterias coronarias (isquemia puede acabar con la muerte de las células del miocardio). Los registros de 12 derivaciones ECG es una manera estándar de diagnosticar una posible isquemia. La isquemia suele apreciarse cuando aparecen cambios en la apariencia de la onda T y en la del intervalo ST.

Hay patrones clásicos de isquemia, los cuales son apreciables solo en una minoría de los eventos isquémicos. En la mayoría de los eventos isquémicos, hay cambios “no

especificados” de la señal ECG, tales como cambios en la onda T que pueden o no causar isquemia,

En un pequeño porcentaje de casos, puede haber isquemia sin cambios evidentes en la señal ECG. Generalmente el ECG no es altamente sensible a la isquemia cardiaca en sí misma, sino que detecta las isquemias cuando la región afectada es lo suficientemente grande (lo que también conlleva una alta tasa de mortalidad cuando se alcanza este punto).

Por ejemplo un paciente con los cambios clásicos en una isquemia aguda es normalmente un buen candidato para una medicación trombolítica. Pacientes sin estos patrones normales no son considerados buenos candidatos para esta medicación porque, de media, el riesgo de aplicar esta medicación supera los posibles beneficios.

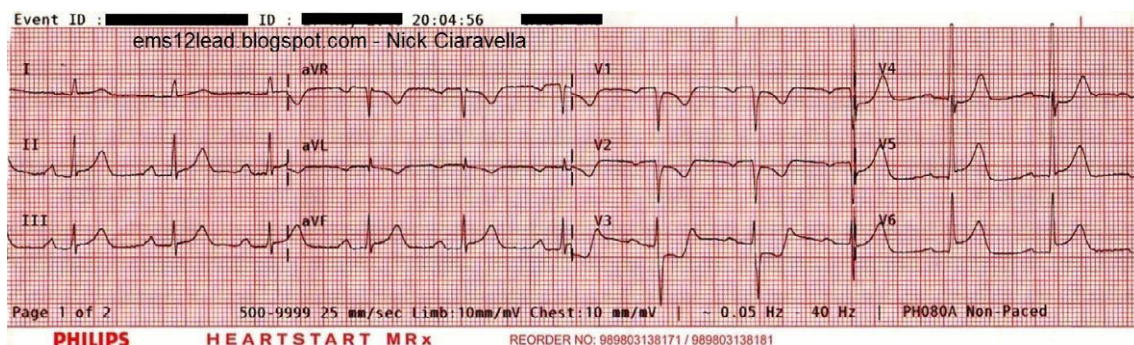


Figura 1.21. Registro 12 derivaciones que muestra un paciente con síntomas isquémicos

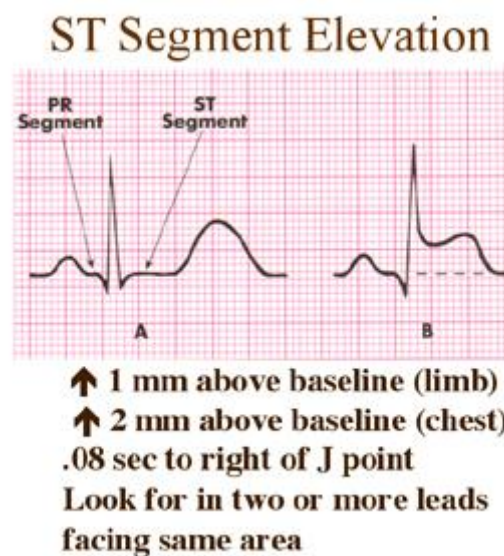


Figura 1.22. Elevación del segmento ST por isquemia cardiaca

1.7 Introducción al análisis de la señal ECG

Para analizar efectivamente un registro clínico de electrocardiograma, es importante usar una aproximación sistemática. A continuación se muestran las principales características que suelen analizarse clínicamente.

- Identificar complejos QRS:
 - o ¿Cuál es la tasa ventricular?
 - o ¿Están los complejos espaciados regularmente?
 - o ¿Son los complejos idénticos en forma para una misma derivación?
- Identificar ondas P:
 - o ¿Existe relación uno a uno entre ondas P y complejos QRS?
 - o ¿Es el intervalo PR de una duración normal?
 - o ¿Cuál es la tasa auricular?
 - o ¿Son las ondas P idénticas en forma para una misma derivación?
- Examinar el complejo QRS en cada derivación:
 - o ¿Es el eje del complejo normal?
 - o ¿Está la amplitud del complejo dentro de los valores normales?
- Examinar los segmentos ST-T:
 - o ¿Hay algún tipo de anormalidad?
 - o Si la hay ¿Sugiere algún tipo de isquemia, infarto o hipotermia?
- Examinar la onda T:
 - o ¿Es su forma normal?
 - o En cada derivación ¿Están orientados en la misma dirección los complejos QRS?
- Examinar el intervalo QT:
 - o ¿Es mayor que la mitad del intervalo RR?

Una vez que una anormalidad es detectada, hay varias posibles explicaciones posibles, muchas de las cuales conducen a varias patologías ECG y puede no ser posible determinar el significado de las anormalidades con certeza. Para confirmar un diagnóstico se buscan otras anormalidades características. Para un individuo dado, comparar un nuevo ECG con un ECG anterior provee de una valiosa información.

Para elaborar este capítulo se han usado principalmente como referencias bibliográficas [5] [6].

2. Características de la señal ECG desde el punto de vista de procesamiento digital de la señal

En este capítulo se va a introducir las características espectrales de la señal ECG que son útiles para una aproximación desde el punto de vista de procesamiento de señales. En estas características me he basado para implementar algunos métodos de análisis de la señal ECG que serán explicados con detalle en siguientes capítulos.

2.1 Análisis espectral de la señal ECG

El contenido espectral a corto plazo para una derivación II de un segmento de señal ECG se muestra en la imagen, contenida en la figura 2.1. Nótese los picos en la densidad de potencia espectral a 1, 4, 7 y 10 Hz respectivamente, que corresponden aproximadamente a la tasa de latidos del corazón (60 bpm), la onda T, la onda P y el complejo QRS respectivamente.

El contenido espectral para cada derivación es bastante similar sea cual sea la configuración, aunque la energía en cada frecuencia puede sufrir algunas variaciones.

En la figura 2.2 se representa la densidad espectral de potencia para una configuración típica (12 derivaciones) de duración 10 segundos. Para realizar una estimación de la similitud espectral entre pares de derivaciones puede calcularse la correlación espectral coherente.

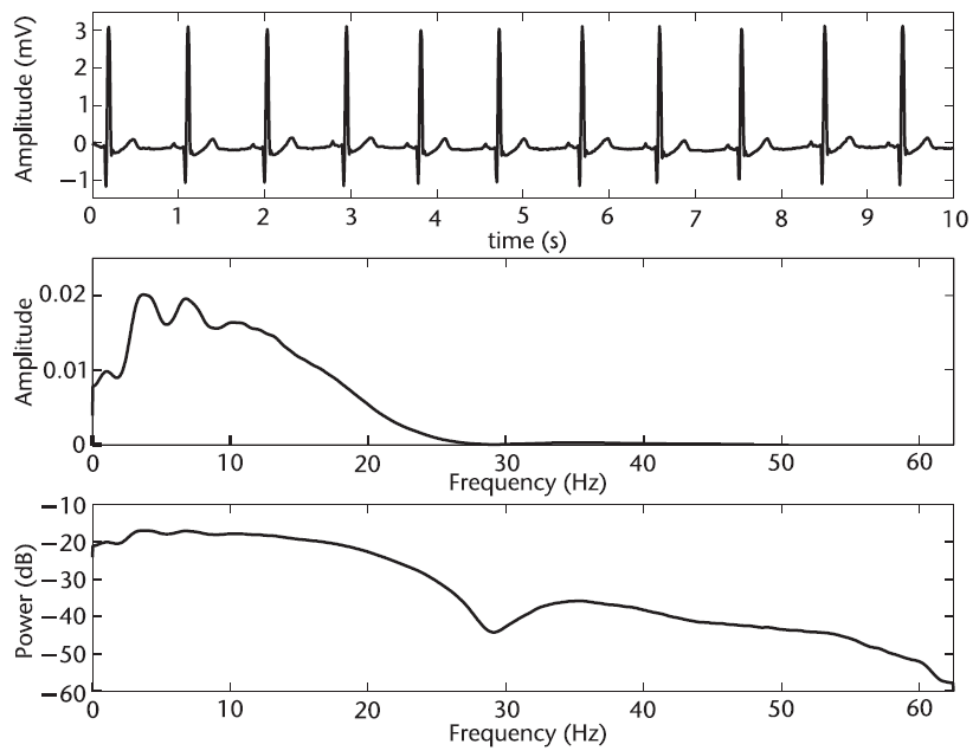


Figura 2.1. Señal ECG 10 segundos junto a su periodograma

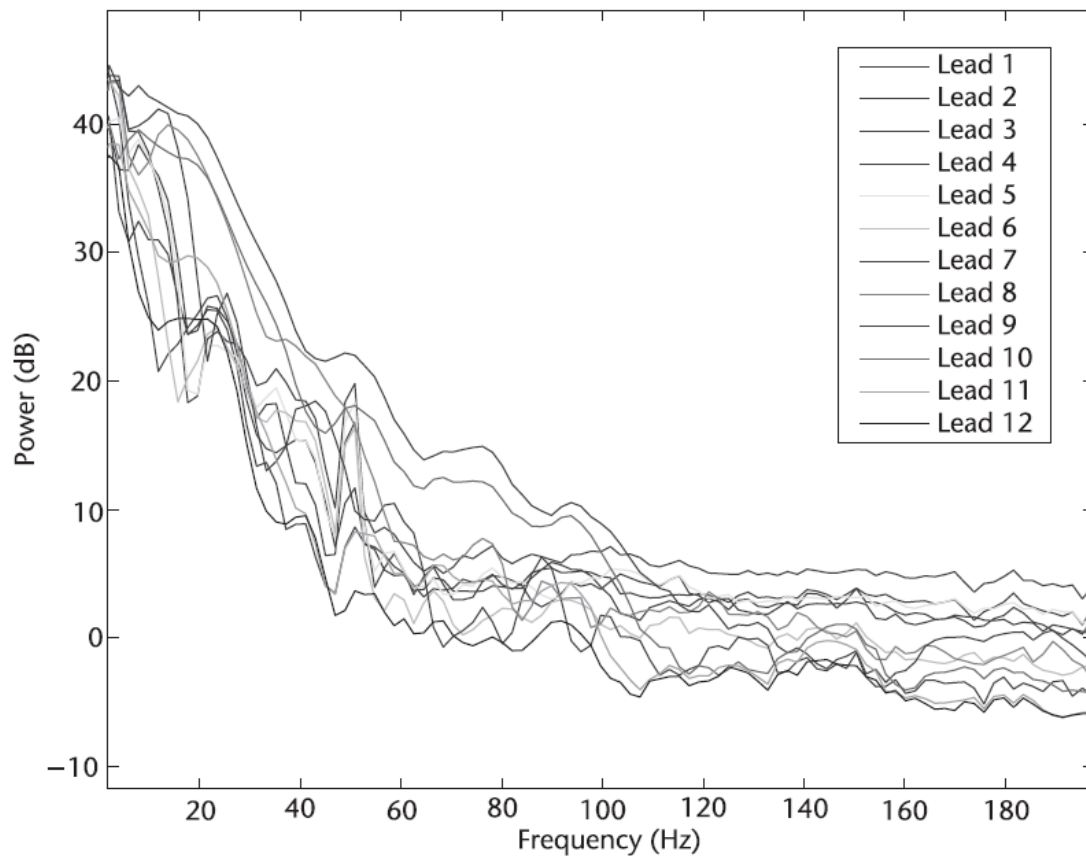


Figura 2.2 PSD (dB/Hz) de 12 derivaciones estándar de una señal ECG de 10 segundos

Siendo la fórmula de la correlación:

$$C_{xy} = \frac{|P_{xy}^2|}{P_x P_y}$$

Donde P_x es la estimación de la potencia espectral de x, P_y es la estimación de la potencia espectral de y e P_{xy} estimación de la potencia espectral cruzada de x e y. Es una función de la frecuencia moviéndose entre los valores 0 y 1 e indica el parecido entre la señal x con la señal y a cada frecuencia.

Esta función dará valores por encima de 0.9 en la mayoría de las frecuencias fisiológicamente importantes (1 a 10 Hz).

Aunque el rango aceptado para el diagnóstico es de 0.05 Hz a los 40 o 100 Hz, sí existe información más allá de esos límites, tanto por arriba como por abajo.

Las arritmias pueden ser agrupadas en arritmias ventriculares y arritmias auriculares. Las ventriculares se manifiestan como grandes distorsiones en la morfología de la onda ya que la despolarización comienza más en los ventrículos que en la aurícula. El complejo QRS se agranda, enmascarando la onda P.

Aunque la ampliación del complejo QRS causa un desplazamiento de sus componentes en el dominio de la frecuencia hacia frecuencias ligeramente más bajas, el espectro es bastante parecido al de un latido normal, con lo que llevar a cabo una clasificación espectral de estas arritmias se torna complicado.

Las arritmias auriculares, en cambio, presentan pequeñas variaciones temporales y de posición de la relativamente pequeña onda P. Por consiguiente son de difícil detección mediante métodos espectrales. A pesar de ello presentan notables variaciones en los tiempos entre latidos, por lo que una forma de detectarlas es realizando análisis estadísticos de intervalos de señal ECG.

2.2 Características clínicas del ECG

Las medidas clínicas del ECG confían en unas magnitudes de fácil medición. Por ejemplo el intervalo entre los latidos y las diferentes amplitudes de las ondas. Para medir correctamente sobre una señal es necesario promediar para evitar las rápidas variaciones entre latidos. Los cambios en la morfología de la señal también pueden ser un indicativo de problemas, sin embargo, un médico puede extraer bastante información de unas pocas y relativamente simples medidas.

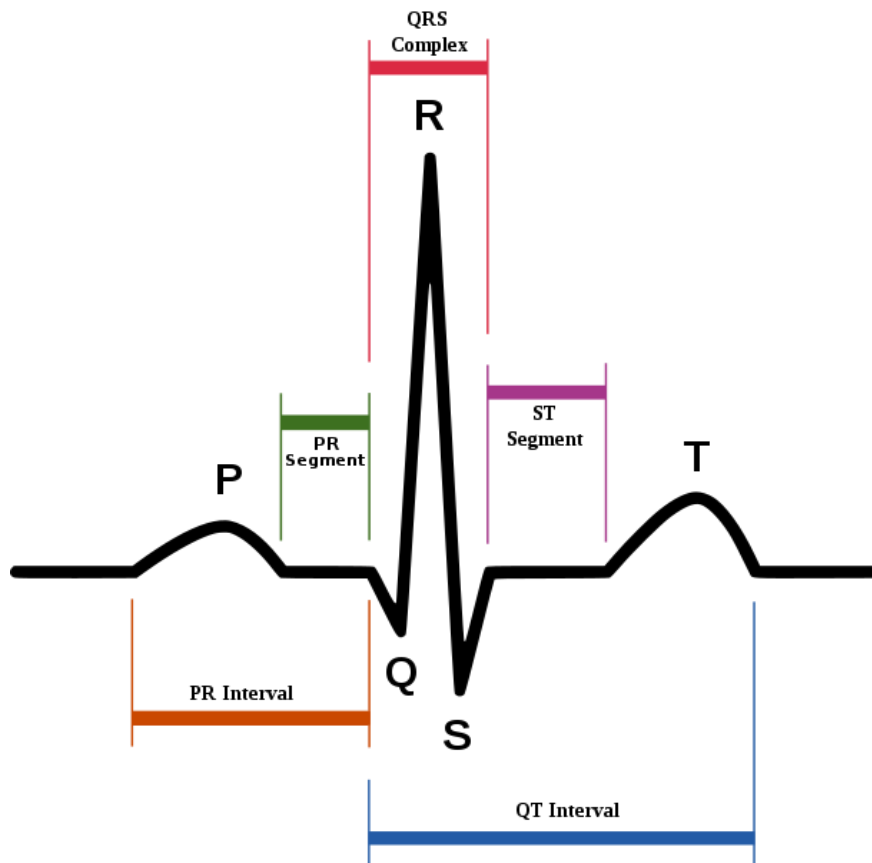


Figura 2.3. Características de una señal ECG estándar

La figura 2.3 representa las características estándar de una señal ECG. Que son a su vez las medidas clínicas comunes que un médico necesita para realizar un diagnóstico. En diagnóstico clínico los latidos por minuto se calculan en intervalos de quince a treinta segundos contando manualmente el número de ondas R en dicho intervalo de tiempo. Desde un punto de vista del procesamiento de señales. La forma habitual es aplicando la siguiente relación:

$$HeartRate = 60 / RRinterval$$

Donde la variable **RRinterval** representa el intervalo en segundos entre ondas R. Hay que tener en cuenta que dicho intervalo (si el paciente no usa un marcapasos) nunca será perfectamente regular por varias causas, una de las más importantes es la influencia de los ciclos respiratorios en el ritmo de los latidos. Para ello es importante, como se indicó anteriormente realizar un promediado en las medidas clínicas.

Generalmente, cada onda e intervalo tiene unos valores considerados normales y un margen de variación fuera del cual puede indicar alguna clase de patología. A continuación se muestra una tabla con los valores más habituales.

Característica	Valor normal	Margen
Ancho onda P	110 ms	± 20 ms
Intervalo PQ/PR	160 ms	± 40 ms
Ancho del complejo QRS	100 ms	± 20 ms
Intervalo QTc	400 ms	± 40 ms
Amplitud onda P	0,15 mV	± 0,05 mV
Altura complejo QRS	1,15 mV	± 0,5 mV
Nivel intervalo ST	0 mV	± 0,1 mV
Amplitud onda T	0,3 mV	± 0,2 mV

Tabla 2.1. Valores normales habituales en ondas ECG

El intervalo PR se extiende desde el comienzo de la onda P hasta el final de la unión del conjunto PQ, al mismo comienzo del complejo QRS (o lo que es lo mismo, el comienzo de la onda R o la onda Q, si ésta existe). Por consiguiente, este intervalo es normalmente conocido como intervalo PQ y representa el tiempo requerido por el impulso eléctrico para viajar desde el nodo SA hasta el ventrículo y los valores normales oscilan entre los 120 y los 200 ms. Está demostrado que el intervalo PR también sufre variaciones (tanto contracciones como expansiones) debido a los intervalos de respiración. Pero son de una menor magnitud y no están totalmente correlados con la señal RR.

El nivel de referencia, respecto al cuál medimos las amplitudes de las diferentes ondas, es el nivel isoelectrico medido durante un corto periodo de tiempo entre la despolarización auricular (onda P) y la despolarización ventricular (complejo QRS). Es el punto de la medida más cercano a los 0 voltios.

El ancho del complejo QRS generalmente sube, o baja, del orden de 1 o 2 milivoltios desde la línea isoelectrica para los latidos normales. Los llamados **artifacts** (errores de medición debidos al movimiento de los electrodos por ejemplo) y los latidos ectópicos pueden ser varias veces más grandes en amplitud. Esto se conoce como **Baseline Wander** y es un problema a tener en cuenta en el procesado de señales de electrocardiograma.

En la siguiente imagen observamos un ejemplo de **baseline wander** obtenido con el software de simulación desarrollado para este proyecto.

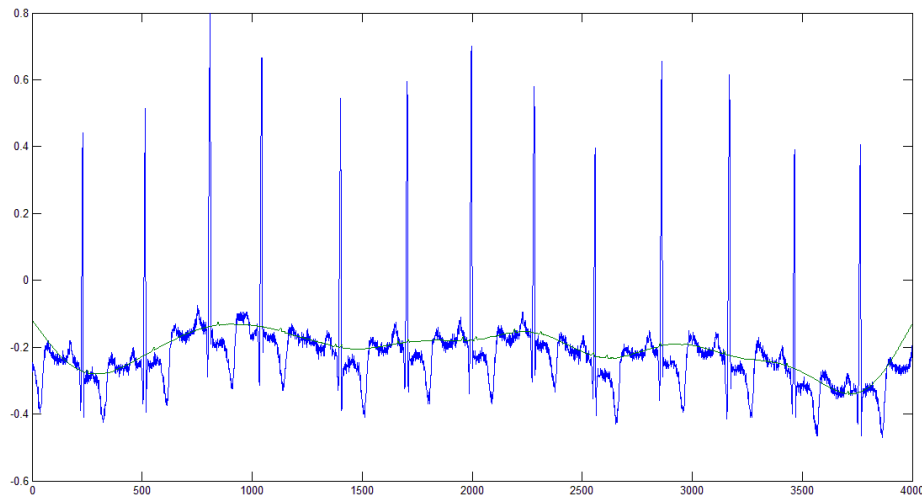


Figura 2.4 Señal ECG con *baseline wander* en azul. En verde el *baseline wander* extraído con filtrado paso bajo

Se puede apreciar que la señal no se encuentra en el nivel de los cero voltios y presenta una oscilación de baja frecuencia. Para garantizar un correcto procesamiento esta oscilación debe eliminarse. Dos métodos suelen usarse, el más habitual es el método de filtrado paso alto (frecuencia de corte de 0,8 Hz en el que desarrollé para este software) o realizar una interpolación polinómica entre los puntos donde se une la onda Q y restar a la señal dicha interpolación.

El punto de inflexión después de la onda S es conocido como el punto J y es a menudo usado para definir el comienzo del segmento ST. En casos normales se espera que sea isoeléctrico porque es la pausa entre la despolarización ventricular y la repolarización. El nivel ST se mide generalmente alrededor de 60 a 80 ms después del punto J.

El intervalo QT se mide entre el comienzo del complejo QRS y el final de la onda T. Se considera que representa el tiempo entre el comienzo de la despolarización ventricular y el fin de la repolarización ventricular, y se considera útil como medida de la duración de la repolarización. La duración del intervalo QT varía dependiendo del ritmo de los latidos, de la edad del paciente y del género. Como otros parámetros de la señal ECG es posible aproximar su dependencia con el ritmo de los latidos multiplicándolo por el siguiente factor α .

$$\alpha = (RR_{average})^{-1/2}$$

Para terminar este subcapítulo se muestran los cambios típicos que se observan cuando se incrementa el ritmo cardíaco [7] [8] [9]

- La media del intervalo RR decrece.
- El segmento PR se acorta y cae hacia abajo (en las derivaciones inferiores)
- La altura de la onda P se incrementa.
- La onda Q se vuelve ligeramente más negativa (sólo cuando los latidos son a un ritmo realmente elevado).
- La anchura del complejo QRS decrece.
- La amplitud de la onda R decrece en las derivaciones laterales solo después un incremento en el ritmo de los latidos.
- La onda S se vuelve más negativa en las derivaciones laterales y en las verticales.
- El punto J se ensancha convirtiéndose en una especie de depresión.
- El nivel ST cambia.
- La amplitud de la onda T se incrementa y se vuelve más simétrica (aunque puede caer inicialmente en el comienzo del incremento de los latidos).
- El intervalo QT se acorta.
- La onda U sí que parece cambiar significativamente, sin embargo las ondas U pueden ser difíciles de identificar debido al corto intervalo entre la onda T y los siguientes latidos de la onda P a alta frecuencia.

2.3 Detección de arritmias

Existen diferentes métodos generales para el análisis de arritmias. Un método es realizar detección de cada complejo QRS y realizar una clasificación de los latidos. Llamándolos o no arritmias en base a las medidas en una serie de estos latidos. Para ello analizamos una sección del ECG que contiene varios latidos, calculamos sus estadísticos (media, varianza, diferencias de potencia a **diferentes frecuencias...**) y en base a ellos se clasifican los latidos. Existen otras técnicas como por ejemplo el estudio de arritmias mediante análisis no lineal pero no será tratado en este proyecto.

Podemos clasificar las arritmias en dos tipos en base al tipo de técnicas de detección de las mismas:

- **Arritmias clasificadas por la categoría de los latidos:** Un conjunto de latidos anormales puede ser clasificado como arritmia. Por consiguiente, la localización de los puntos fiduciales y una condición que limite el número de latidos considerados normales es suficiente para detectar estas arritmias. Las arritmias supraventriculares y la presencia sostenida de latidos ectópicos supraventriculares a una tasa superior a cien latidos minuto son arritmias detectables por estos métodos.
- **Arritmias clasificadas por análisis en frecuencia:** A veces no hay ningún punto fiducial consistente en la señal ECG, lo que implica que el análisis habitual es imposible. En estos casos se suelen analizar los cambios en la frecuencia que suceden durante las arritmias.[10] [11]

2.4 Ruido y artifacts en la señal ECG

Gran parte de los problemas que las técnicas de procesado digital de señales tienen que solucionar se deben a que las señales biológicas siempre llegan contaminadas con algún tipo de ruido. Ya sea este ruido biológico (por ejemplo la actividad eléctrica producida por los músculos esqueléticos) o artificial (las interferencias causadas por la alimentación 50/60 Hz). Las señales indeseadas que pueden alterar nuestras medidas son las siguientes:

- **Interferencias causadas por la alimentación (50/60 Hz):** Suele ser del orden de los 50/60 Hz con un margen de $\pm 0,2$ Hz.
- **Ruido de contacto:** La pérdida de contacto entre los electrodos y la piel.
- **Artifacts debidos al movimiento del paciente:** Los movimientos del paciente provocan variaciones de la impedancia entre los electrodos y la piel causando cambios en la señal ECG.
- **Ruido electromiográfico:** Actividad eléctrica debido a la contracción de los músculos esqueléticos.
- **Baseline Wander:** Usualmente causado por la respiración.
- **Artifacts debidos al hardware:** Tales como la saturación de los amplificadores, ruido de los diferentes subsistemas...
- **Ruido electroquirúrgico:** Generado por otros aparatos médicos.
- **Artifacts debido al procesado de la señal:** Como por ejemplo las oscilaciones de Gibbs*.

(*) *Efecto Gibbs: Se denomina efecto Gibbs al efecto rizado que aparece en la transformada de Fourier de funciones que presentan discontinuidades.* [12]

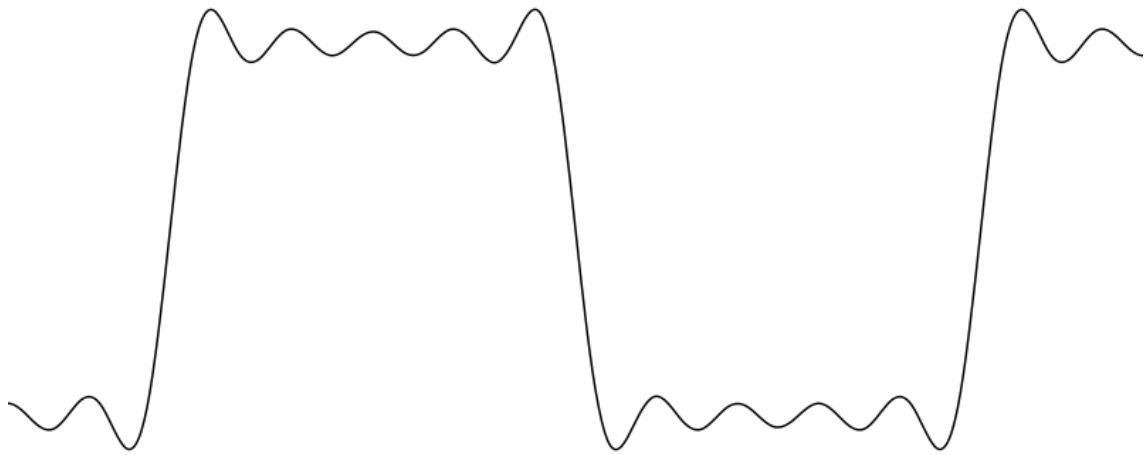


Figura 2.5. Imagen del efecto Gibbs para la transformada de Fourier de la función escalón

3. Descripción de los algoritmos

Este capítulo servirá para realizar una explicación de los diferentes algoritmos programados en este proyecto para llevar a cabo el correcto procesamiento de las señales de ECG. No solo se describirán los dos algoritmos codificados propiamente dichos para realizar la detección de los complejos QRS, sino también se describirán los filtros de preprocesado que el usuario puede activar en función de sus necesidades.

Filtros de preprocesado

Para adecuar la señal de ECG a los requerimientos, es necesario filtrar previamente la señal para asegurar que llegue lo más limpia posible a los algoritmos aplicados. Los filtros implementados en el software son:

- Filtro paso alto para eliminar ***Baseline Wander***.
- Filtro paso bajo para extraer dicho ***Baseline Wander***.
- Filtros Notch 50/60 Hz
- Filtro adaptativo 50/60 Hz

3.1 Filtro paso alto para eliminar Baseline Wander

El efecto conocido como ***baseline wander*** presenta frecuencias bastante bajas. Según [13] puede elegirse como frecuencia de corte 0,8 Hz para un correcto eliminado.

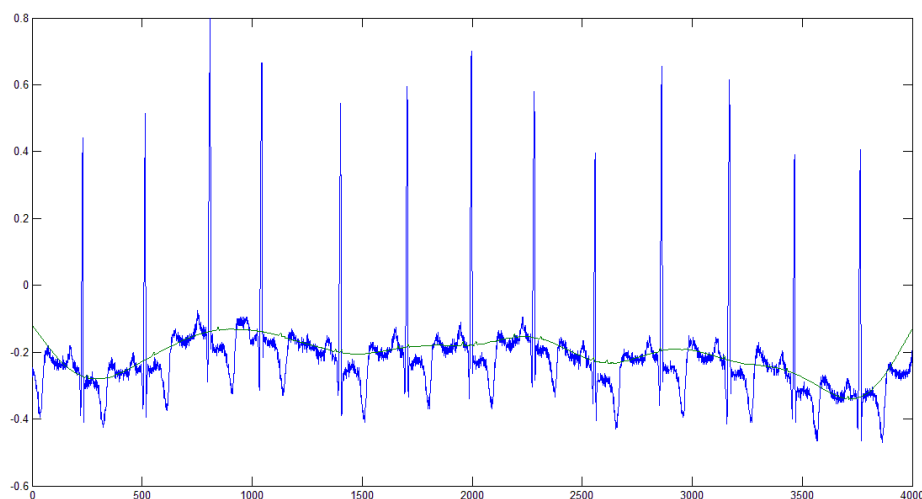


Figura 3.1. Ejemplo de señal ECG con Baseline wander

La implementación del filtro se encuentra en la página 73.

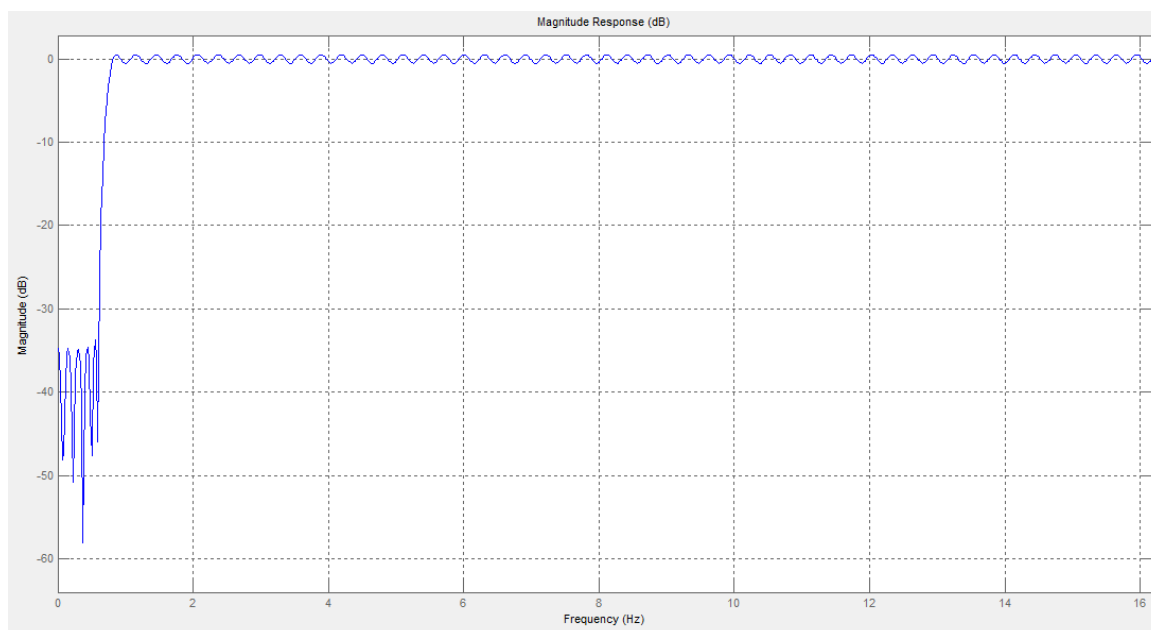


Figura 3.2. Densidad espectral de potencia del filtro paso alto

Se puede apreciar de forma clara como la frecuencia de corte es realmente baja. Consistiendo para casi todas las frecuencias prácticamente un filtro paso todo. Este filtro es muy para el procesamiento, pero particularmente para calcular el ancho del complejo QRS por lo cual siempre debe estar activo si se desea realizar mediciones sobre la anchura de dichos complejos.

3.2 Filtro paso bajo para extraer el Baseline Wander

A pesar de ser una señal indeseada que provoca problemas, en ciertas ocasiones el *baseline wander* puede resultar de utilidad. Si se extrae y se procesa convenientemente esta señal puede utilizarse en algoritmos denominados EDR o *ECG Derived Respiration* que pueden obtener una estimación de la frecuencia respiratoria del paciente.

Para obtener dicha señal debemos usar un filtro paso bajo contrario al anterior filtro expuesto. Dicho filtro tendrá una frecuencia de corte de 0,8 Hz y prácticamente solo dejara pasar la componente del *baseline wander*. La implementación del filtro junto con las de los demás filtros se encuentra en la página 73 del anexo que contiene el código fuente.

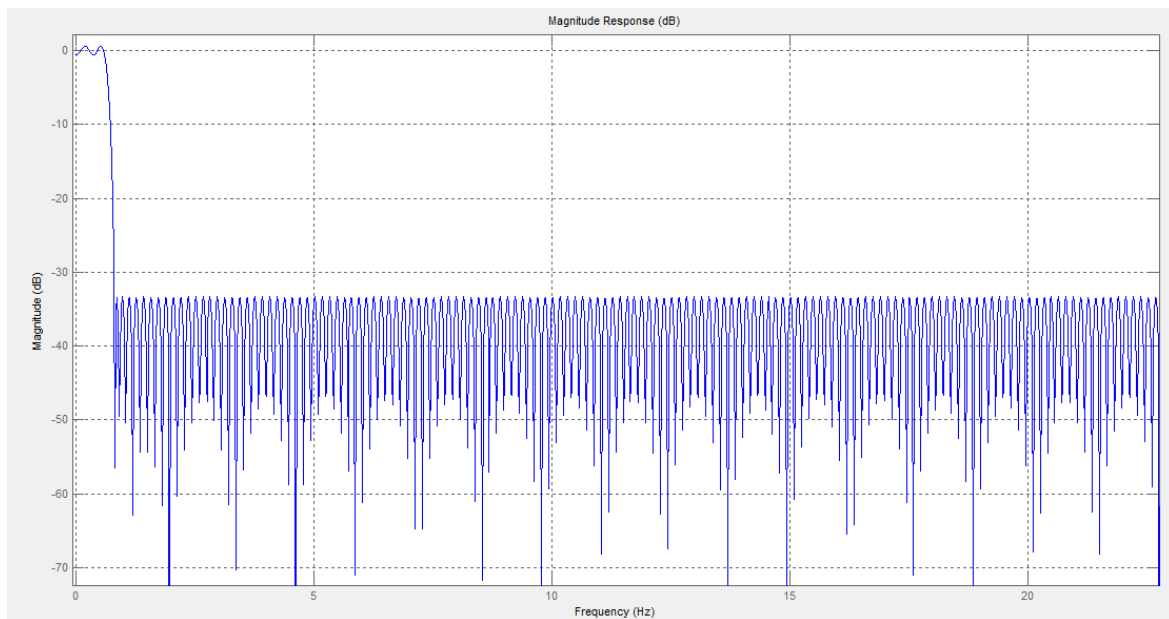


Figura 3.3. Densidad espectral de potencia filtro paso bajo

3.3 Filtros Notch 50/60 Hz

Las señales provenientes de la alimentación de los equipos deben ser eliminadas antes de realizar cualquier procesamiento sobre ellas. Dependiendo del país de origen, la alimentación introduce interferencias a los 50 o 60 Hz. Para eliminar estas interferencias se han seguido dos aproximaciones en este proyecto, esta primera realiza un filtrado lineal banda eliminada 50/60 Hz a elección del usuario.

El código de los filtros está en la página 73 del anexo:

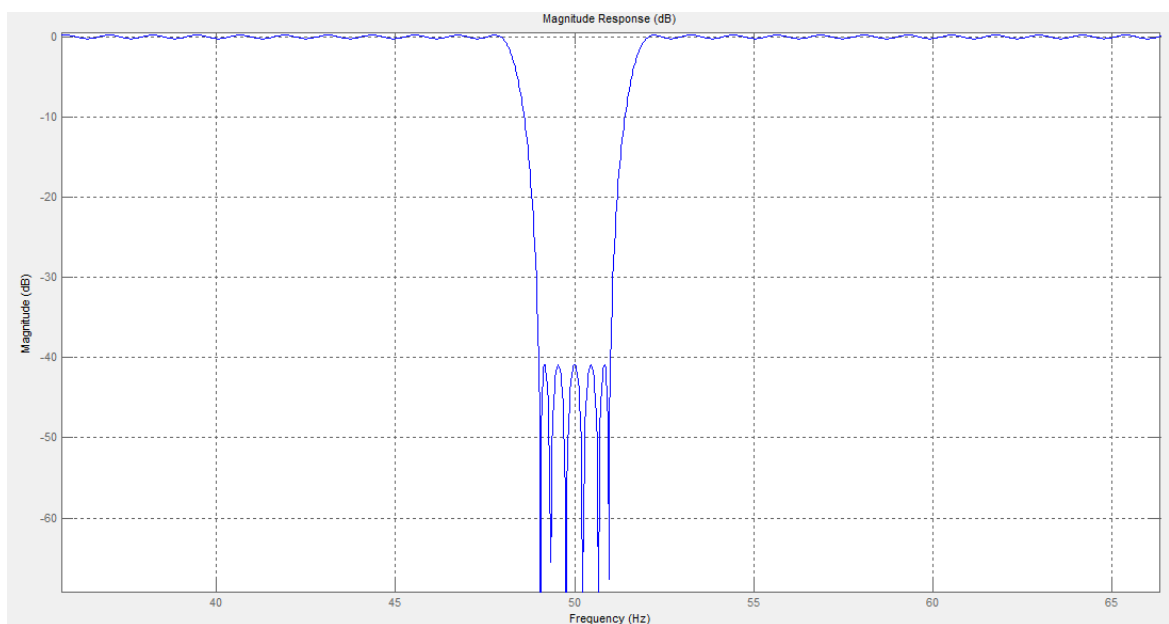


Figura 3.4. Densidad espectral de potencia filtro notch 50 Hz

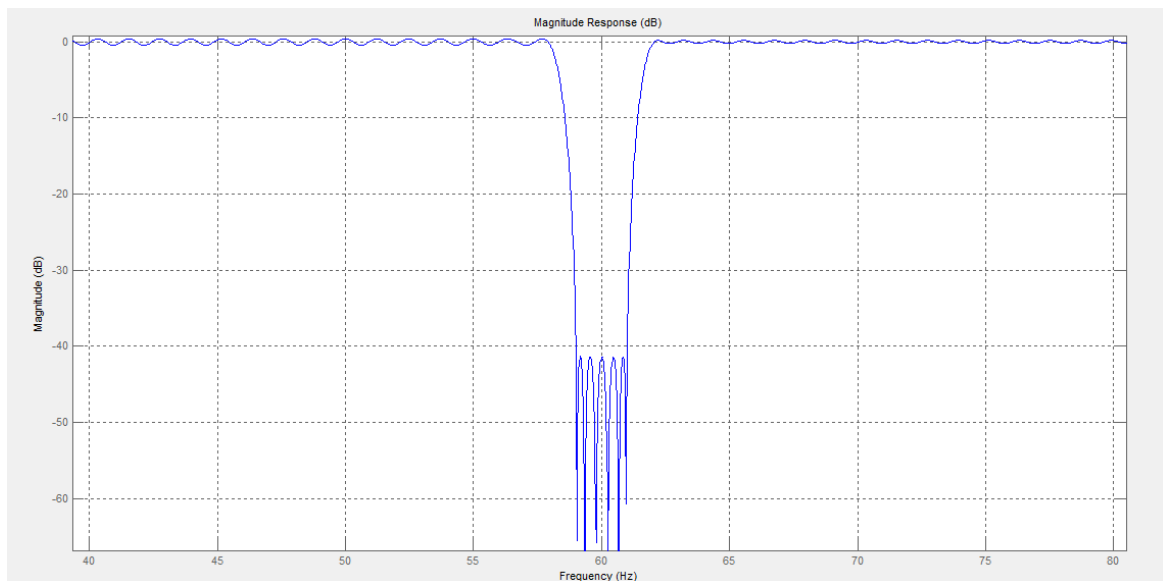


Figura 3.5. Densidad espectral de potencia notch 60 Hz

3.4 Filtro adaptativo 50/60 Hz

Para resolver el problema de las interferencias causadas por la alimentación otra posible aproximación es realizar mediante un esquema de filtrado adaptativo. Se ha usado como algoritmo adaptativo el *Least Mean Squares (LMS)*

La implementación en Matlab del filtro es la siguiente:

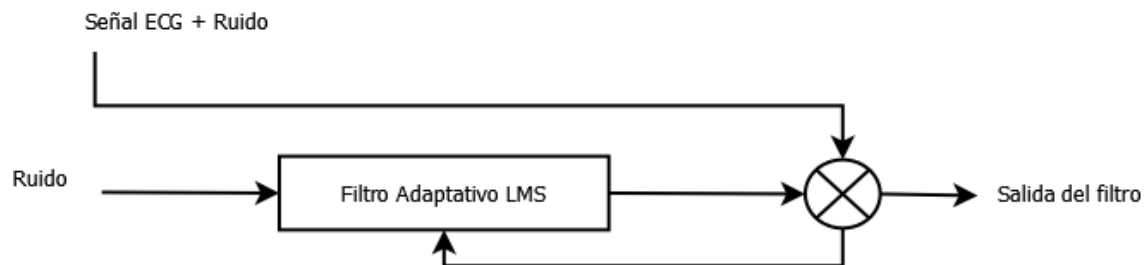


Figura 3.6 Típico esquema de cancelación de ruido mediante filtrado adaptativo

El código Matlab que implementa dicho esquema se encuentra en la página 82 del anexo.

3.5 Nociones comunes sobre los algoritmos detección QRS:

Antes de describir los dos algoritmos empleados en este proyecto se debe diferenciar las características necesarias en un detector QRS. Un detector QRS está compuesto de dos bloques, uno llamado preprocesador y otro de decisión [14].

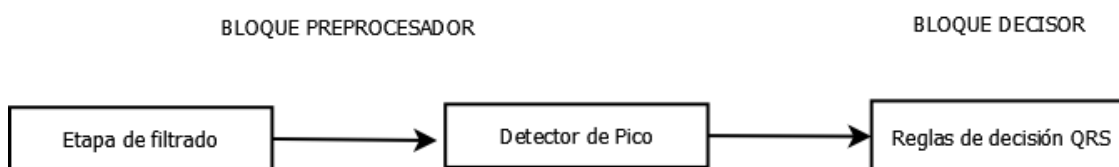


Figura 3.7. Esquema básico de un detector QRS

En el bloque preprocesador se suelen realizar una o varias etapas de filtrado (lineal o no lineal) para simplificar la señal ECG a un tipo de señal con menos irregularidades. Una vez que la señal ha sido convenientemente tratada por los filtros se aplica un detector de pico que devuelve los posibles picos de la señal (indicativos de onda R). Para finalizar, el bloque decisor debe implementar una serie de reglas basadas en investigaciones anteriores por las cuales es capaz de discriminar entre posibles falsos positivos y positivos.

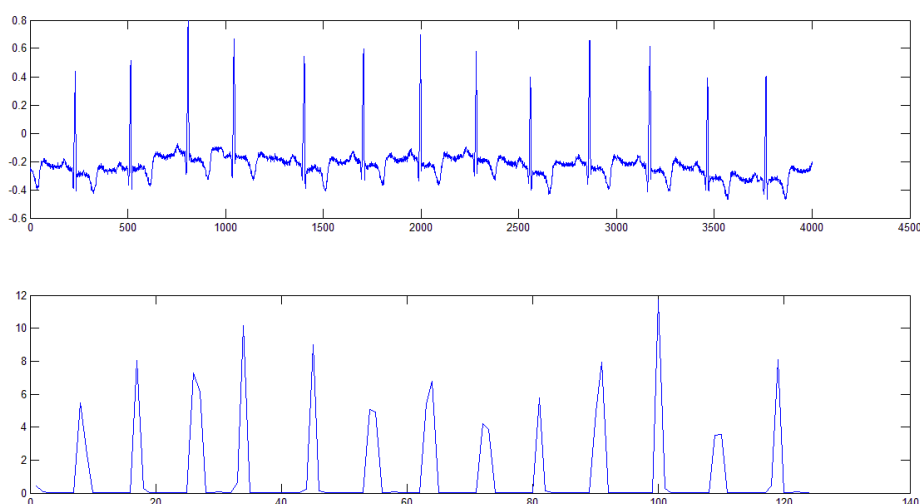


Figura 3.8. Arriba señal ECG, abajo señal ECG que ha pasado por una etapa de filtrado del preprocesador

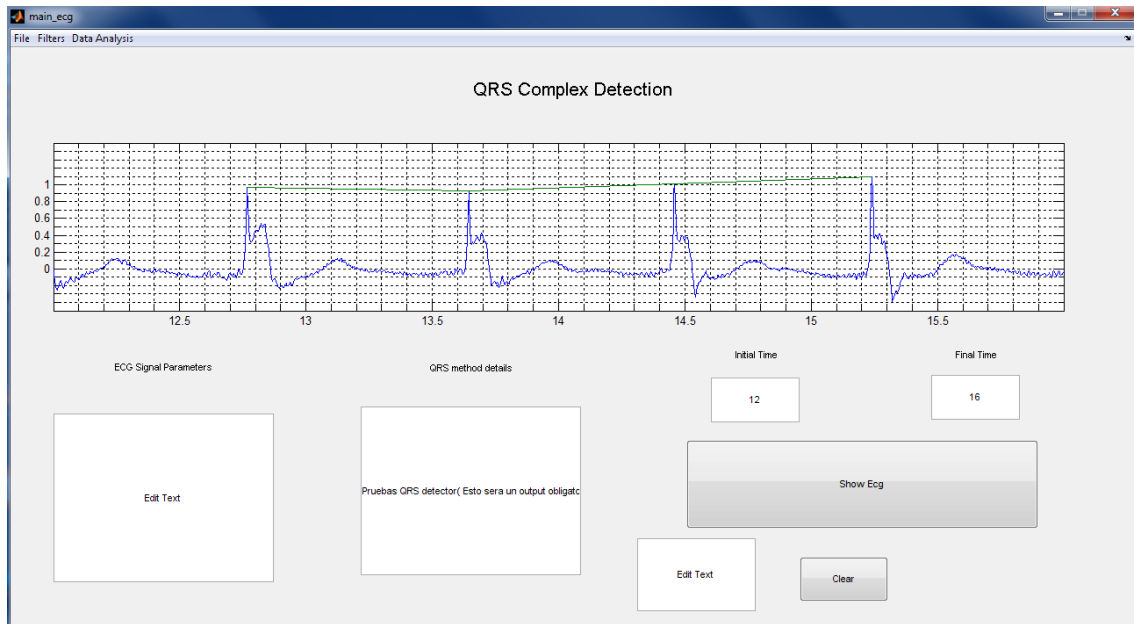


Figura 3.9. Resultado final del algoritmo gracias a las reglas de decisión

3.6 Detector QRS con etapas de filtrado lineal y no lineal

El primer algoritmo implementado en este proyecto se basa en el artículo de Patrick S Hamilton y Willis J Tompkins *Quantitative investigation of QRS detection rules* [1]. Implementa como etapa de preprocesado un conjunto de filtros de distintos tipos definidos en [15]. Finalmente implementa un detector de pico y un conjunto de reglas de detección QRS. El diagrama de bloques de la etapa de filtrado es el siguiente:

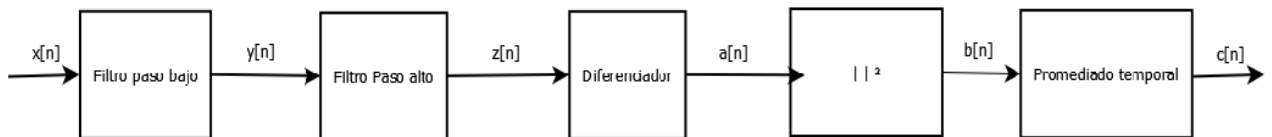


Figura 3.10. Diagrama de bloques de la etapa de filtrado

El filtro paso bajo obedece a la siguiente ecuación en diferencias:

$$y[n] = 2y[n - 1] - y[n - 2] + x[n] - 2x[n - 6] + x[n - 12]$$

Que presenta la siguiente respuesta en frecuencia:

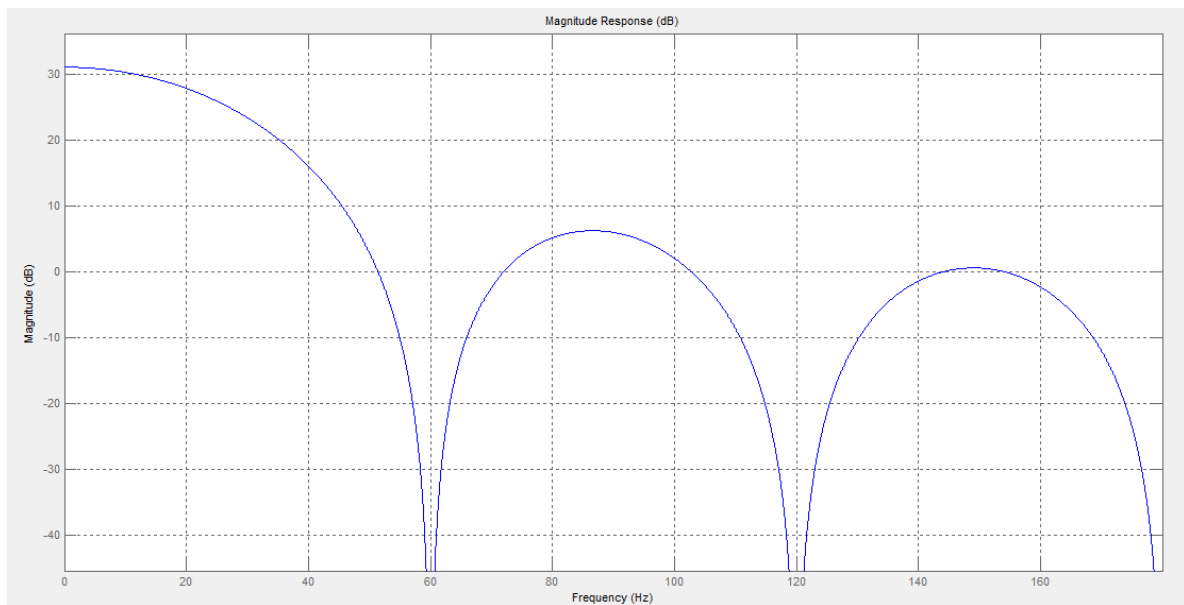


Figura 3.11. Respuesta en frecuencia del filtro paso bajo

Su diagrama de polos y ceros:

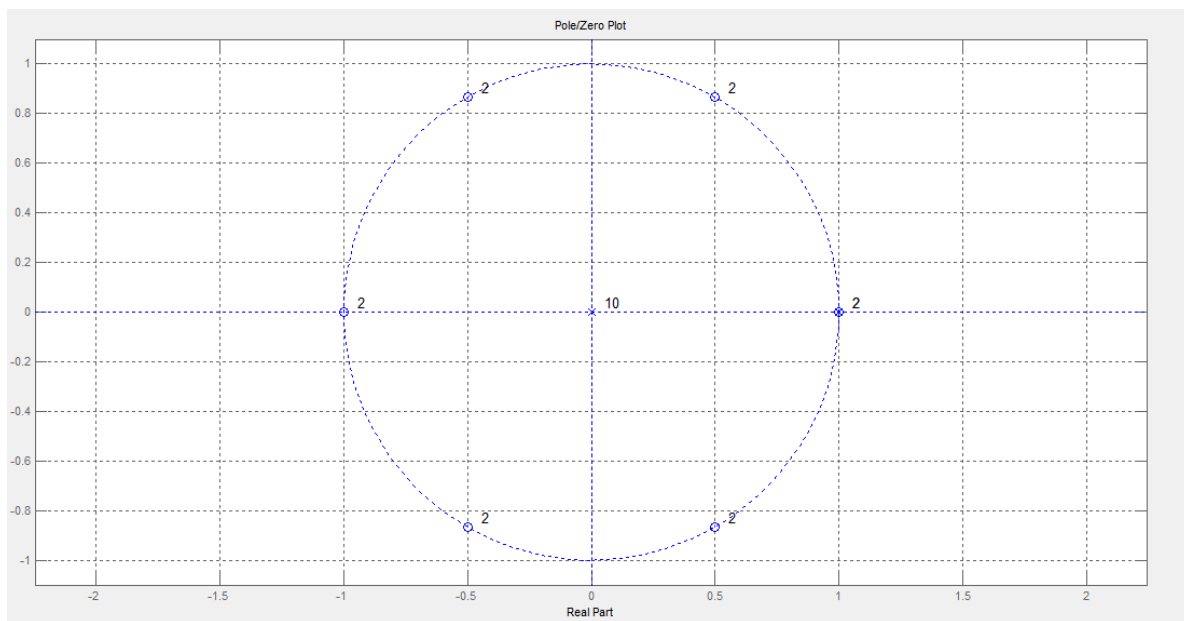


Figura 3.12. Diagrama de polos y ceros del filtro paso bajo

El filtro paso alto está implementado con la siguiente ecuación:

$$z[n] = z[n - 1] - \frac{y[n]}{32} + y[n - 16] - y[n - 17] + \frac{y[n - 32]}{32}$$

Presenta la siguiente respuesta en frecuencia:

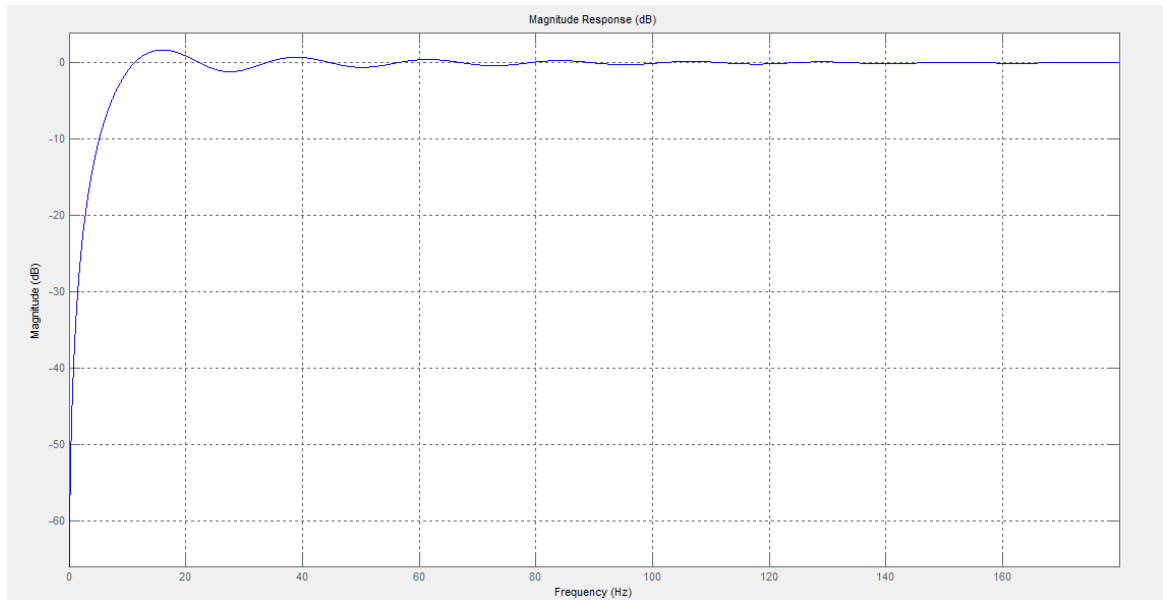


Figura 3.13. Respuesta en frecuencia del filtro paso alto

Su diagrama de polos y ceros es:

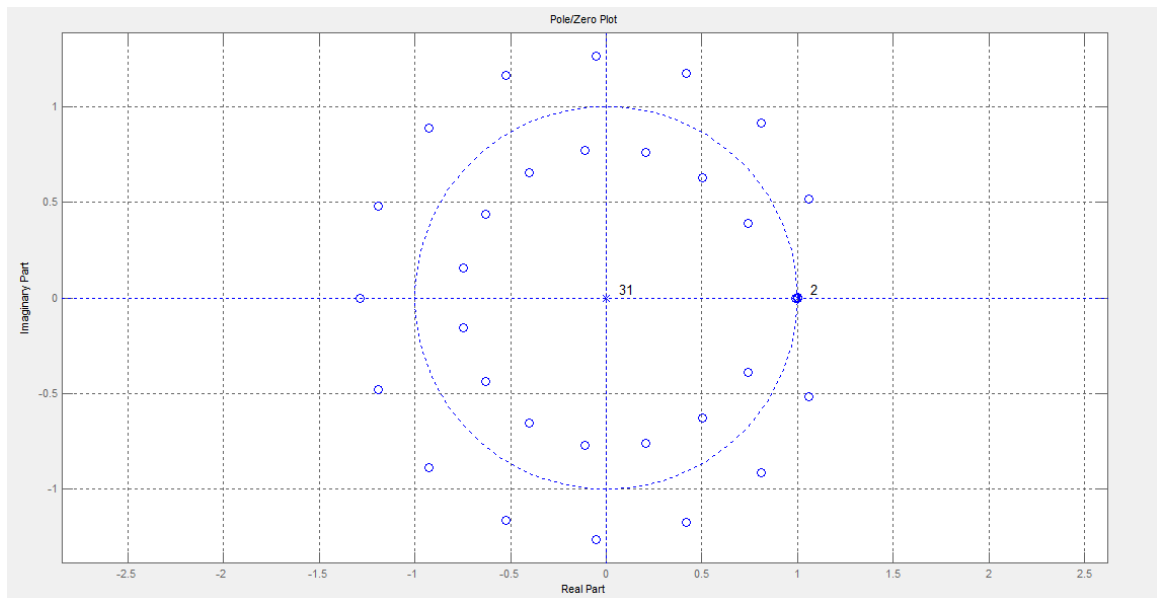


Figura 3.14. Diagrama de polos y ceros del filtro paso alto

La ecuación del diferenciador es:

$$a[n] = \frac{2z[n] + z[n - 1] - z[n - 3] - 2z[n - 4]}{8}$$

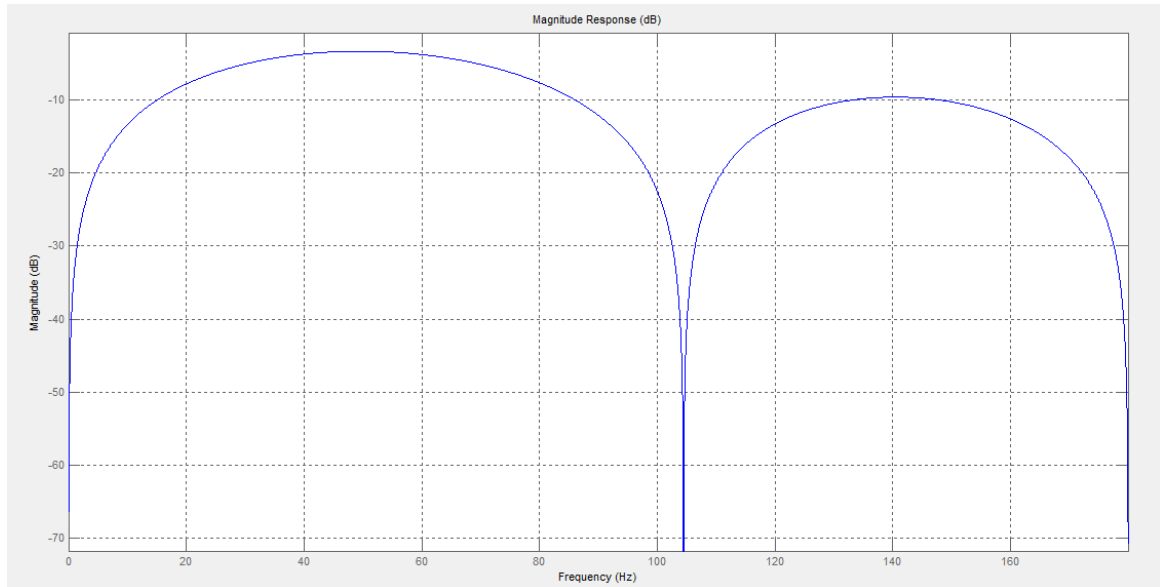


Figura 3.15 Respuesta en frecuencia del diferenciador

Presenta el siguiente diagrama de polos y ceros:

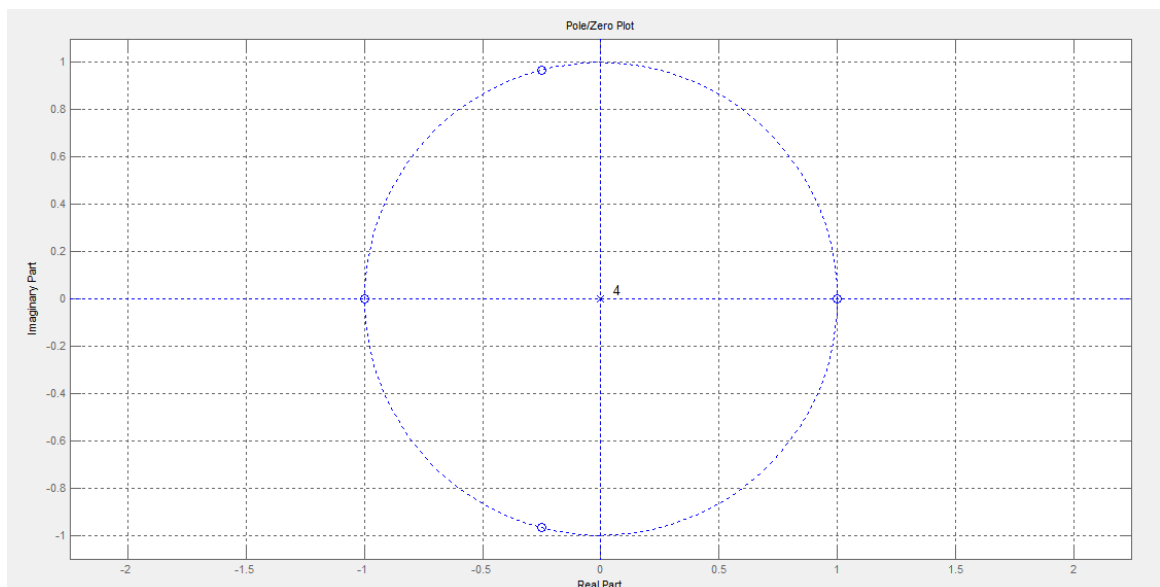


Figura 3.16 Diagrama de polos y ceros del diferenciador

A la salida del diferenciador se realiza un promediado cuadrático temporal con las 32 muestras más recientes:

$$\frac{1}{32} \sum_{1}^{32} |b[n]|^2$$

Una vez finalizada la etapa de procesamiento mediante filtros es necesario aplicar un detector de pico para localizar las posibles ondas R. El detector de pico implementado sigue el siguiente esquema:

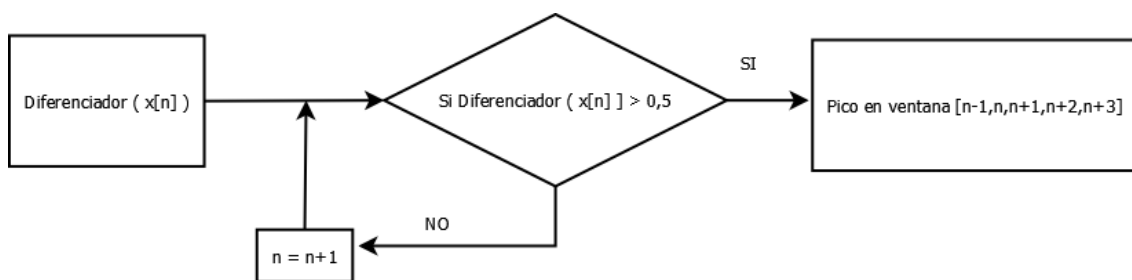


Figura 3.17. Algoritmo del detector de pico

El detector de pico realiza la ecuación en diferencias:

$$y[n] = x[n] - x[n - 1]$$

Que corresponde a la diferencia de primer orden de la salida que otorga la etapa de filtrado. Una vez realizada analiza en una ventana de cinco muestras cual es la mayor y a la mayor muestra la marca como positivo. Para los tiempos de muestreo de las señales de las bases de datos usadas. La ventana se corresponde con un tiempo de 0.0112s. La condición de 0,5 fue determinada empíricamente usando un conjunto de muestras de señales ECG.

Hay que tener en cuenta que los puntos candidatos para ser máximos del complejo QRS no lo son sobre la señal original, sino sobre la señal anteriormente filtrada y procesada. Esto implica la necesidad de tener en cuenta que como mínimo los máximos del detector de pico llevan un desfase de 32 muestras con respecto a la señal real. Por ello el detector QRS simplemente aplica en torno a los máximos detectados por el detector de pico una ventana de 64 muestras en la señal real (el doble por seguridad). Sobre las muestras de esta ventana se detecta el máximo y el punto que sea detectado como máximo será finalmente catalogado como máximo definitivo.

El código de los diferentes subsistemas se encuentra en el anexo, a partir de la página 77.

3.7 Detector QRS con preprocesador basado en filtrado paso banda y transformada Hilbert

Para este proyecto se ha realizado un algoritmo original para la detección de complejo QRS. La principal motivación ha sido el uso de un menor número de filtros para lograr una mayor eficiencia computacional.

El algoritmo usa una etapa de preprocesado diferente. Para aislar la componente espectral del complejo QRS se necesita, tal como se demuestra en [16] un filtro paso banda con una frecuencia central de 17 Hz.

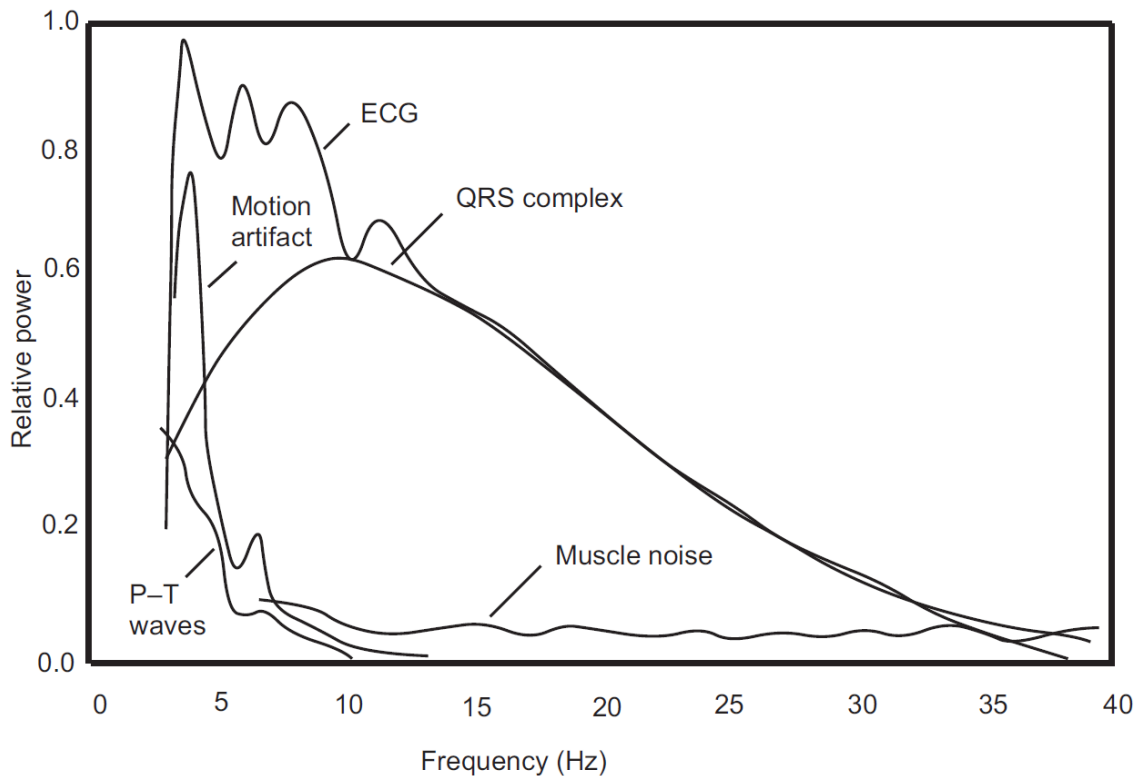


Figura 3.18. Estimación espectral de las diferentes ondas que componen la señal ECG

El diagrama del sistema completo es el siguiente:

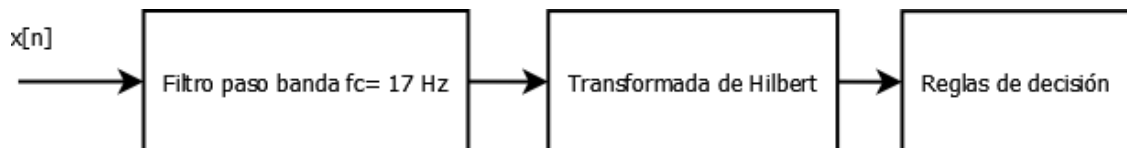


Figura 3.19. Detector QRS con filtrado paso banda y transformada Hilbert

La salida del filtro paso aísla las componentes espectrales fundamentales del complejo QRS. El siguiente paso es realizar una transformada Hilbert [17].

Una de las aplicaciones prácticas de la transformada Hilbert es delimitar la envolvente de una señal. Matemáticamente se puede expresar como:

$$\text{Envolvente de } f(t) = |\text{HilbertTransform}[f(t)]|$$

lo que devuelve la envolvente de la señal sobre la que se ha aplicado la transformada de Hilbert. Esto permite suavizar la señal para que las reglas de detección sean más precisas.

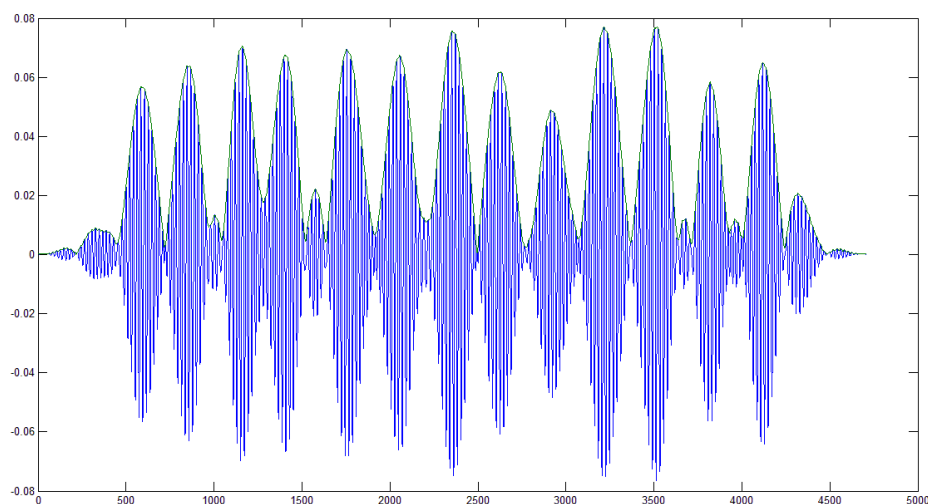


Figura 3.20. Ejemplo de transformada de Hilbert sobre señal ECG filtrada paso banda.

El decisor en este caso es algo distinto al implementado en el anterior algoritmo.

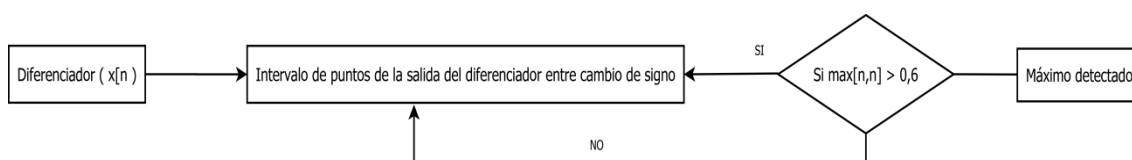


Figura 3.21. Detector de pico

El decisor devuelve los picos de la onda R que sirven para localizar los complejos QRS. El código implementado se encuentra en el anexo, página 81.

4. Comparativa entre los diferentes algoritmos

En este capítulo se va a realizar una comparativa entre los dos algoritmos descritos en el capítulo anterior. Se realizarán dos comparaciones, la primera orientada a la tasa de aciertos de los dos algoritmos y la segunda orientada a la comparación del tiempo de ejecución de los distintos algoritmos.

Usaremos la notación algoritmo 1 para denominar al detector QRS con etapas de filtrado lineal y no lineal y algoritmo 2 para el detector QRS con preprocesador basado en filtrado paso banda y transformada Hilbert.

4.1 Tasa de aciertos

Para comprobar la efectividad de los algoritmos de este proyecto, se han realizado pruebas con las bases de datos de señales biológicas de <http://www.physionet.org/>. Se han usado los conjuntos de registros de arritmias.

El test se realizó con diez de los registros de 30 minutos cada uno. Obteniéndose los siguientes resultados:

	Algoritmo 1 (tasa de acierto)	Algoritmo2 (tasa de acierto)
Registro 100	98,02%	92,32%
Registro 101	97,07%	90,86%
Registro 102	97,93%	91,96%
Registro 103	95,91%	91,56%
Registro 104	96,12%	92,34%
Registro 105	98,73%	94,22%
Registro 106	92,34%	88,76%
Registro 107	94,51%	89,97%
Registro 108	96,23%	91,27%
Registro 109	97,21%	93,19%

Tabla 4.1. Tasas de éxito de los dos algoritmos

El algoritmo 1 presenta una mejor tasa de acierto que el algoritmo 2. El enfoque desarrollado en primer algoritmo no logra mejorar el rendimiento del algoritmo. También se ha observado que la etapa de filtrado es mucho más robusta a la hora de suavizar la señal que la combinación filtrado paso banda y obtención de la envolvente mediante transformada de Hilbert.

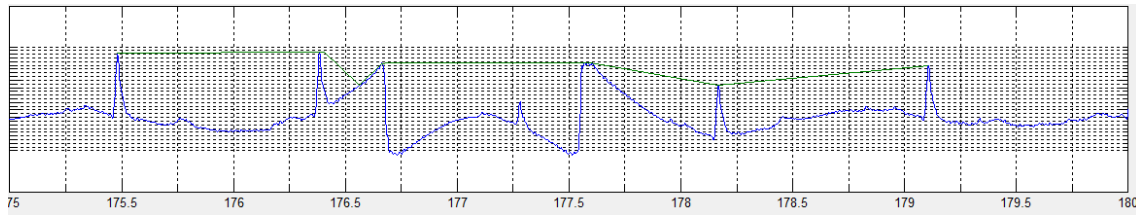


Figura 4.1. Ejemplo de falso positivos. Las grandes perturbaciones de la señal afectan la fiabilidad de los algoritmos

4.2 Eficiencia computacional

Actualmente existen varios proyectos para realizar plataformas de adquisición de ECG de muy bajo coste tanto para labores humanitarias como para proyectos educativos. Muchos de estos proyectos suelen realizarse en la plataforma Arduino, que provee una arquitectura "hardware open source" [18].

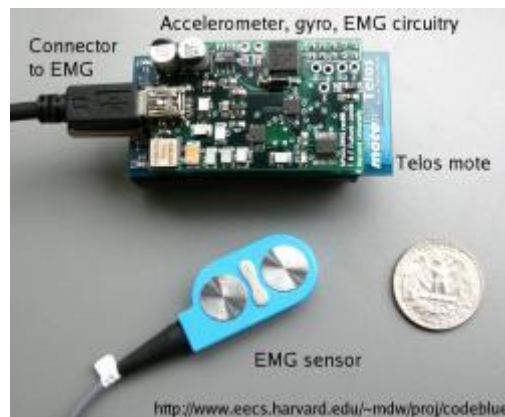


Figura 4.2. Ejemplo de proyecto educativo con hardware open source

Para estos proyectos, que suelen contar de fondos limitados es de importancia la máxima eficiencia de los algoritmos empleados. Unos algoritmos bien optimizados pueden resultar a largo plazo en un ahorro de energía.

Además, en los aparatos denominados Holters, los cuales sirven para la grabación de larga duración de señales ECG es necesario. Los registros suelen realizarse durante unas 24 horas.

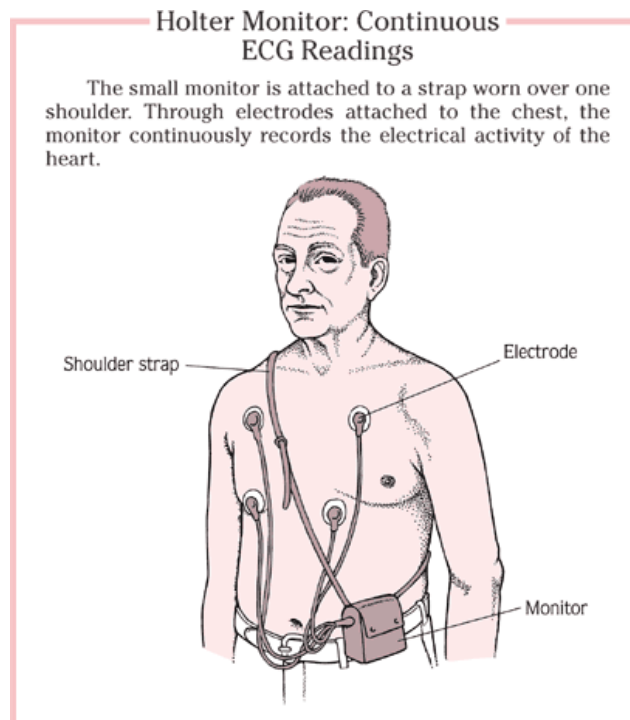


Figura 4.3. El paciente debe llevar los electrodos durante al menos 24 horas

En el siguiente test se ha medido el tiempo medio que la CPU ha tardado para computar las señales de 30 minutos de duración. Se usaron los mismos registros que en el caso anterior. Los resultados se muestran a continuación:

	Algoritmo 1 (segundos)	Algoritmo2 (segundos)
Registro 100	2,17	1,49
Registro 101	2,23	1,78
Registro 102	2,37	1,90
Registro 103	2,20	1,51
Registro 104	2,12	1,54
Registro 105	2,23	1,68
Registro 106	2,19	1,48
Registro 107	2,16	1,51
Registro 108	2,26	1,80
Registro 109	2,19	1,40

Tabla 4.2. Tiempos medios de ejecución de los dos algoritmos

Tiempo medio de ejecución algoritmo 1: 2,212s

Tiempo medio de ejecución algoritmo2: 1,609s

El algoritmo 2 mejora en un 27% el tiempo computacional. A pesar de su menor tasa de acierto, en un entorno donde prime el ahorro el algoritmo 2 puede proveer una mejor solución.

4.3 Sumario

Para este proyecto se ha desarrollado un algoritmo, desde cero, para la detección de complejos QRS. El objetivo de dicho algoritmo es lograr una tasa aceptable de aciertos con un coste computacional menor que otras alternativas más estudiadas.

Se ha comparado **la eficiencia de esta propuesta con un algoritmo “clásico” desarrollado en [1]**. En el test de tasa de aciertos el algoritmo propuesto ha presentado un comportamiento, con unas tasas de aciertos cercanas a las logradas por Hamilton y Tompkins en [1].

En el test de tiempo de ejecución el algoritmo propuesto logra una mejora promedio de un 27% en el tiempo de ejecución con respecto a [1]. Lo que en entorno de escasos recursos, tanto hardware como energéticos puede conducir a un importante ahorro a largo plazo.

Se puede afirmar que el algoritmo propuesto puede ser más adecuado para entornos donde **los dispositivos deben ser “low-cost” o donde el ahorro energético es importante.**

5. Descripción de la herramienta software

La herramienta software desarrollada permite realizar un análisis sencillo de señales ECG obtenidas principalmente de la base de datos de Physionet. El software está abierto a la implementación de nuevos algoritmos por parte del usuario. La implementación realizada lleva como default dos algoritmos. Uno desarrollado por Hamilton y Tompkins [1] y otro desarrollado expresamente para este proyecto, ambos algoritmos están detallados en el capítulo anterior. El software ha sido desarrollado en Matlab.

5.1 Menú Principal

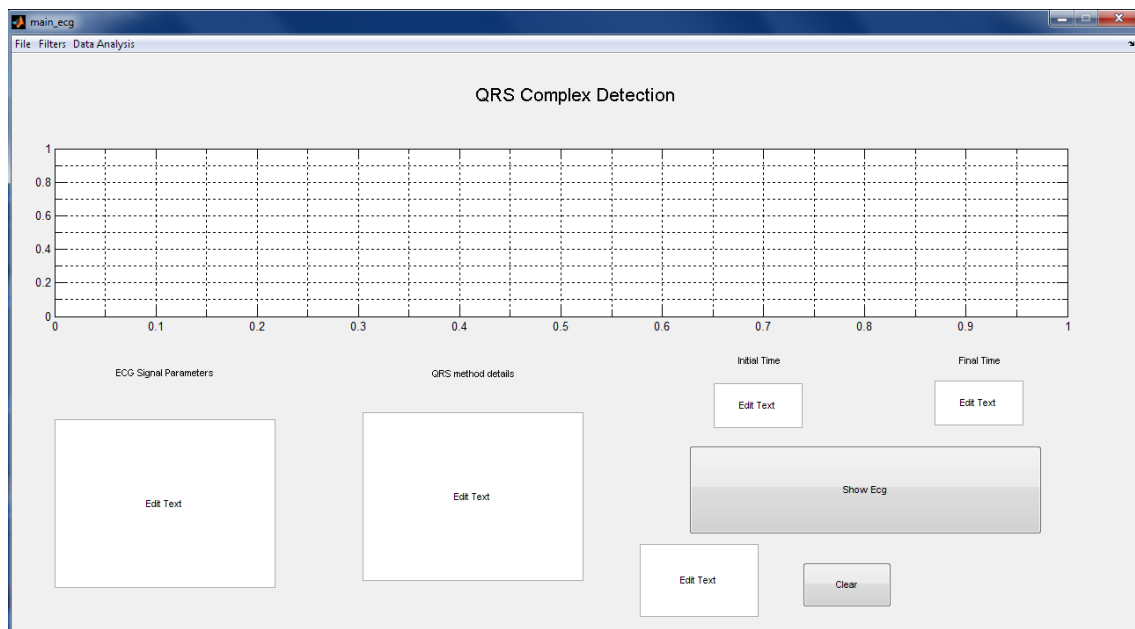


Figura 5.1. Pantalla principal

El menú principal es la primera interfaz que tiene el usuario final con la herramienta. Por defecto muestra una gráfica en la cual el eje X representa el tiempo y el eje Y representa la amplitud de la señal. Las escalas de estos ejes variaran con los valores que el usuario introduzca en los campos *Initial Time* y *Final Time*. Dichos campos delimitan el rango en el cuál el software analizará la señal.

Una vez cargada la señal deseada (File -> Load Ecg Signal) se debe seleccionar el método deseado para analizar la señal.

5.2 Desarrollo de nuevos métodos

Desarrollar nuevos algoritmos para el software es un proceso simple. El usuario debe tener en cuenta unos aspectos:

El código que implemente el usuario debe estar contenido en una función con el encabezado que sigue:

```
function [R_value,points]=QRS_detector(ecg,tsamp)
```

Donde los valores de entrada son:

Ecg : la señal ECG en un formato válido

tsamp: el tiempo de muestreo de la señal en segundos

Los valores de salida son:

R_value: Valores de pico de las ondas R detectadas

Points: Array con los puntos donde se han encontrado ondas R

Para cargar un método durante la simulación en el menú File seleccionamos Load QRS Method y pinchamos en el archivo .m que hemos generado.

Además el programador debe tener en cuenta que el software ya posee una serie de filtros para preprocesar la señal, con lo que el algoritmo que se diseñe no debe incluirlos. Los filtros son los siguientes:

- Filtro paso alto para eliminar el *baseline wander* *
- Filtro paso bajo por si es necesario la extracción de dicho *baseline wander**
- Filtros Notch de 50/60 Hz
- Filtro adaptativo para eliminar la señal de 50 Hz

5.3 Ejemplo de funcionamiento

Una vez seleccionado el algoritmo deseado se debe presionar el botón Show ECG para mostrar los resultados del algoritmo. Además de la representación gráfica que se muestra en pantalla se generan archivos de texto con todos los valores analizados para un análisis numérico con herramientas adecuadas como el mismo Matlab o Microsoft Excel.

Algunos ejemplos de funcionamiento:

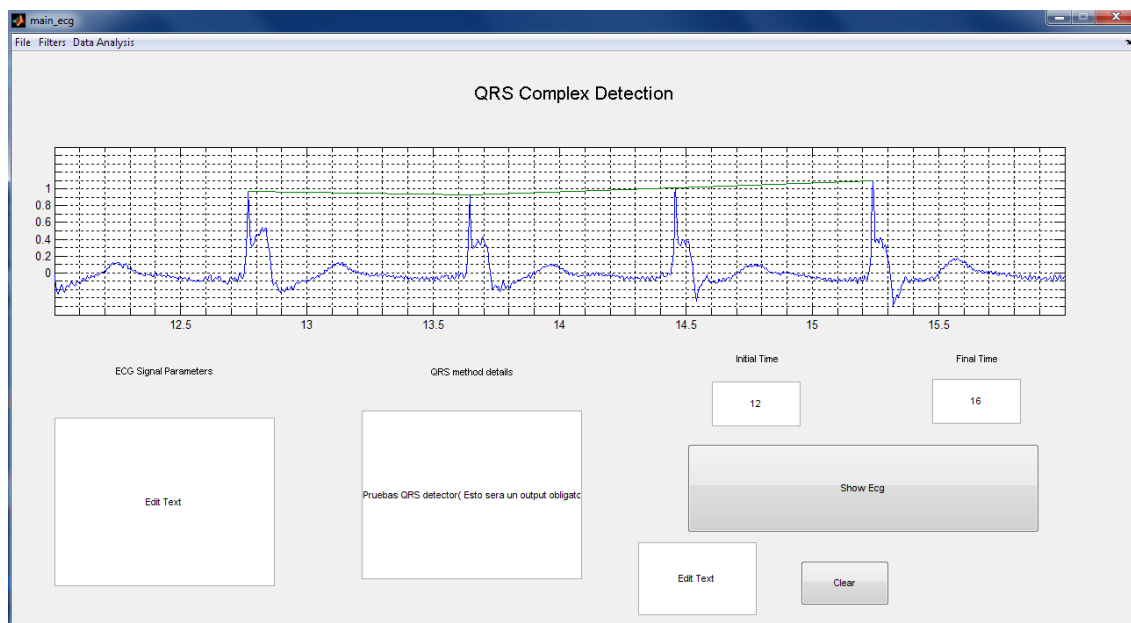


Figura 5.2. Señal analizada entre los segundos 12 y 16

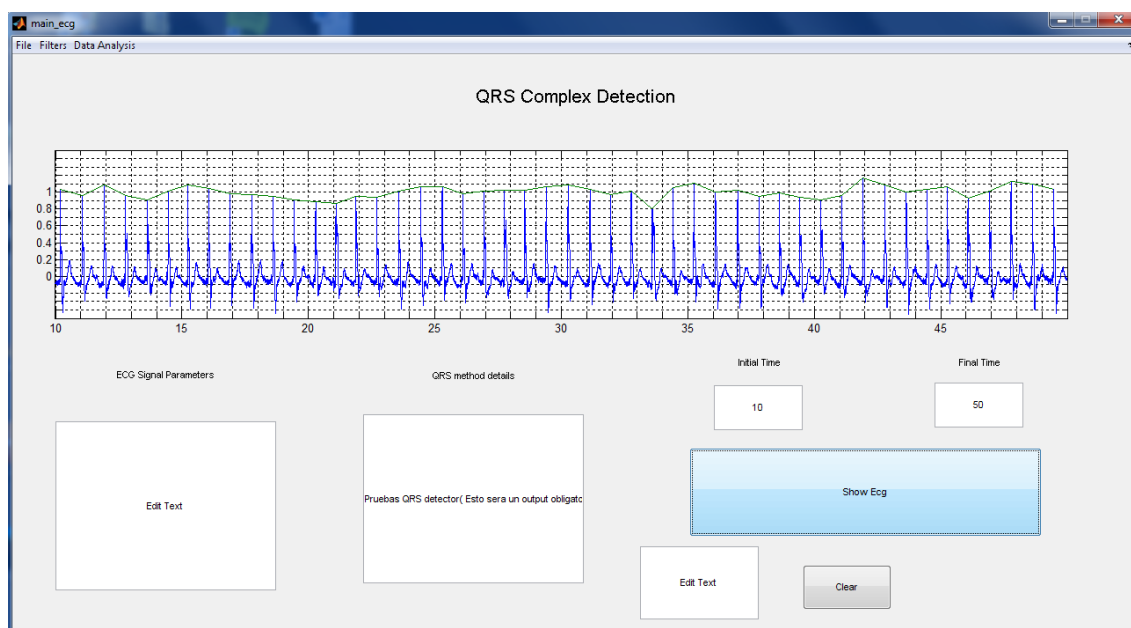


Figura 5.3. Señal analizada entre los segundos 10 y 50

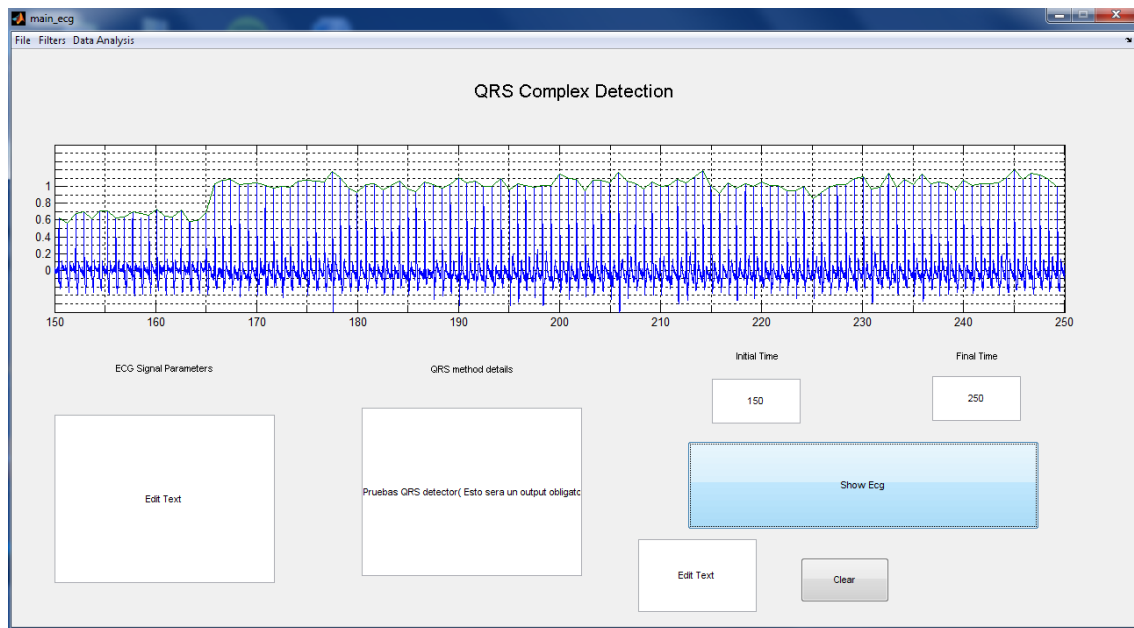


Figura 5.4. Señal analizada entre los segundos 150 y 250

Esta interfaz permite el análisis de tramos de interés que permitan determinar la eficacia del algoritmo. Se pueden hacer análisis más específicos con las herramientas de análisis de datos.

5.4 Análisis de datos

Se pueden realizar análisis de media y varianza de las siguientes magnitudes:

- Heart Rate variability (Variabilidad de los latidos)
- Valor de la onda R
- Valor de la anchura del complejo QRS.

Estas medidas permiten detectar fragmentos de la señal que potencialmente pueden ser sinónimo de arritmia cardíaca. Algunos ejemplos:

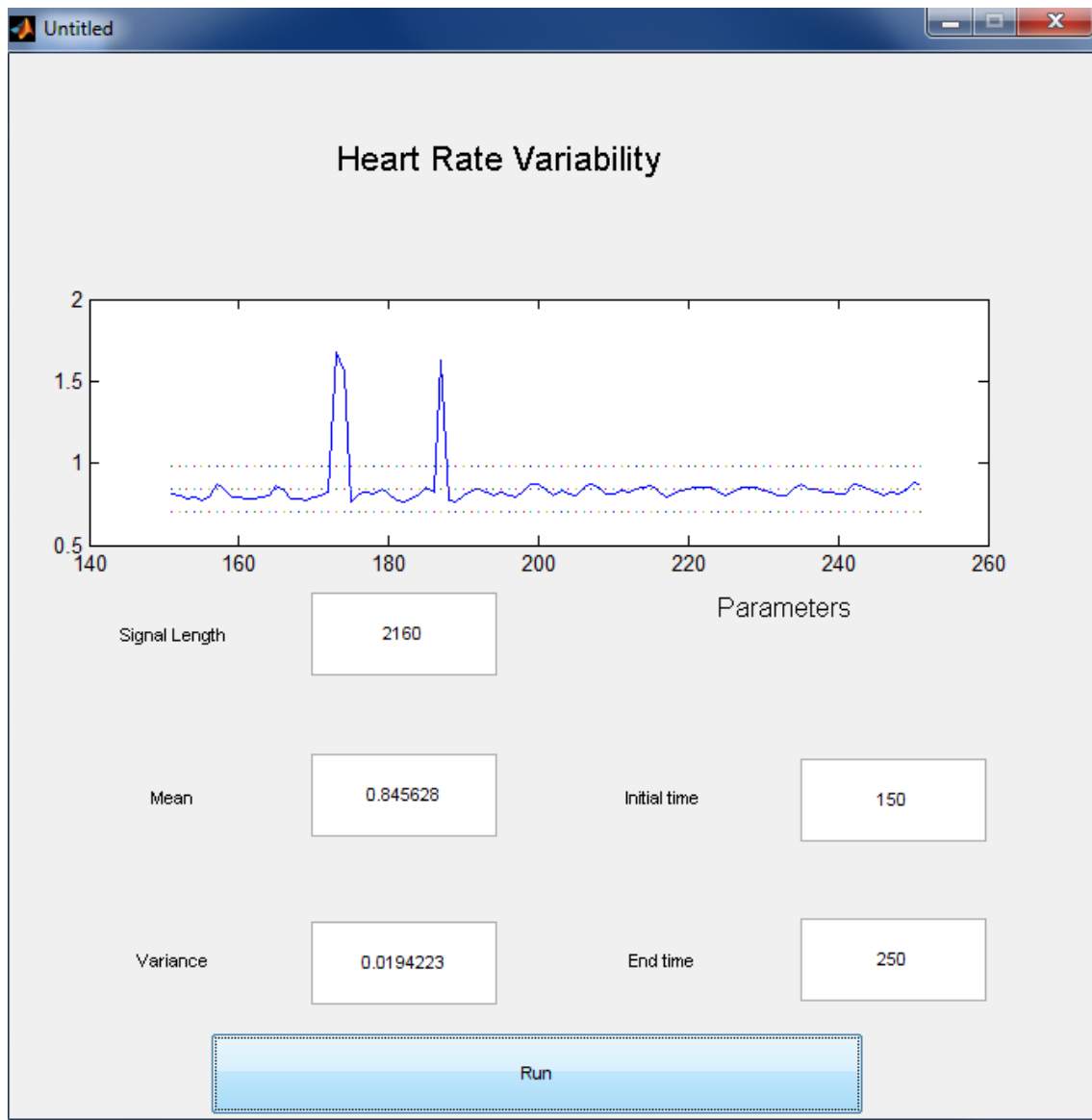


Figura 5.5. Herramienta para medir la variabilidad de los latidos

Introduciendo los límites de tiempos deseados se obtiene una representación gráfica acompañado de los valores numéricos.

Otros ejemplos esta vez con la amplitud de la onda R:

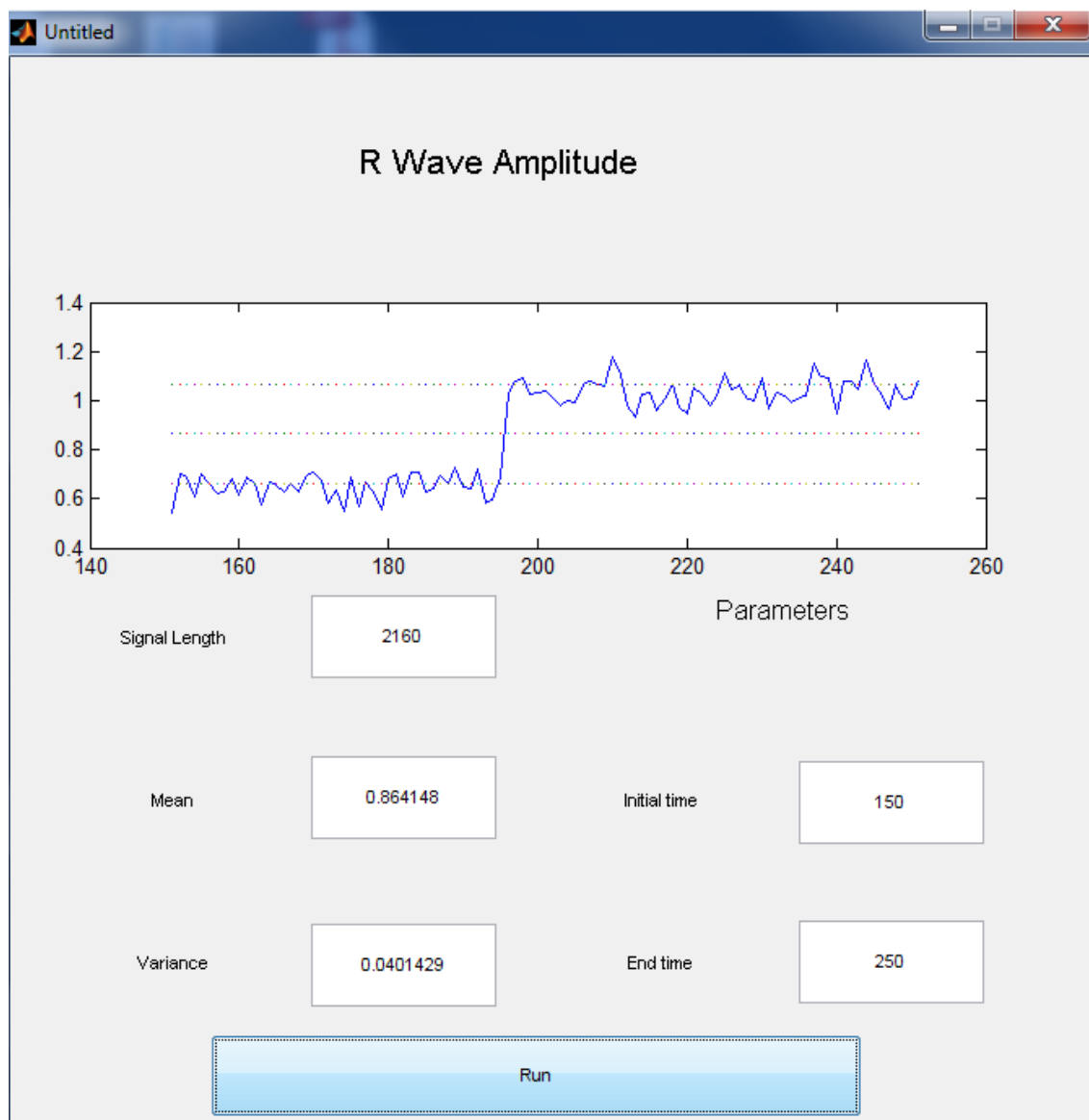


Figura 5.6. Herramienta para medir la amplitud de la onda R

5.5 Logs:

En casos de señales de muy larga longitud, es conveniente tener datos numéricos para poder analizarlos con las herramientas adecuadas. Es por ello que la herramienta genera logs de cada simulación. Los logs muestran en un archivo de texto todos los datos brutos que la simulación pueda ofrecer. Se encuentran disponibles en la carpeta llamada Logs.

Ejemplo de archivo de texto con los datos de una simulación

Time (s)	R wave (V)	QRS (s)	RR interval (s)	Heart Rate (bpm)
0.296800	1.016140	0.005600	0.868000	69.124424
1.164800	0.983957	0.014000	0.814800	73.637703
1.979600	1.028795	0.011200	0.806400	74.404762
2.786000	0.964262	0.011200	0.837200	71.667463
3.623200	0.856344	0.103600	0.792400	75.719334
4.415600	1.029829	0.005600	0.887600	67.598017
5.303200	1.197302	0.103600	0.789600	75.987842
6.092800	1.017936	0.005600	0.879200	68.243858
6.972000	0.965142	0.005600	0.910000	65.934066
7.882000	1.026849	0.123200	0.733600	81.788441
8.615600	1.144536	0.005600	0.798000	75.187970
9.413600	0.937069	0.014000	0.879200	68.243858
10.292800	1.033603	0.061600	0.792400	75.719334
11.085200	0.950945	0.011200	0.862400	69.573284
11.947600	1.108315	0.014000	0.820400	73.135056
12.768000	0.967735	0.030800	0.904400	66.342326
13.672400	0.865217	0.014000	0.842800	71.191267
14.515200	1.052528	0.005600	0.747600	80.256822
15.262800	1.072875	0.005600	0.840000	71.428571
16.102800	1.030781	0.005600	0.862400	69.573284
16.965200	0.973622	0.056000	0.842800	71.191267
17.808000	0.986231	0.005600	0.809200	74.147306
18.617200	0.935382	0.005600	0.929600	64.543890
19.546800	0.899930	0.131600	0.789600	75.987842
20.336400	0.905057	0.014000	0.798000	75.187970
21.134400	0.878482	0.014000	0.750400	79.957356
21.884800	0.937254	0.005600	0.929600	64.543890
22.814400	0.921252	0.047600	0.778400	77.081192
23.592800	1.007452	0.016800	0.898800	66.755674
24.491600	1.099737	0.011200	0.817600	73.385519
25.309200	1.032901	0.005600	0.828800	72.393822
26.138000	0.965478	0.008400	0.879200	68.243858

6. Presupuesto

Al ser este proyecto principalmente un desarrollo software, los conceptos a tener en cuenta están mucho más limitados que en otra clase de proyecto que engloben más recursos físicos.

Concepto	Coste unitario	Unidades	Precio total
Ordenador personal	800 €	1	800 €
Software Matlab	2100 € licencia básica*	1	2100 €
Mano de obra	30€/hora	120 horas	4600 €

(*) Para el proyecto se usaron las licencias de Matlab que posee la Universidad Politécnica de Madrid. Para un desarrollo profesional la licencia debería ser adquirida de manera legal.

7. Conclusiones y futuras líneas de investigación y desarrollo

El proyecto ha tenido principalmente las siguientes líneas de trabajo:

- Estudio de las señales ECG, tanto su origen biológico como sus características matemáticas.
- Propuesta de un método de procesamiento digital del ECG para la detección del complejo QRS y su comparación con un método de detección clásico [1].
- Desarrollo de un algoritmo que implementa ambos métodos: el procesamiento propuesto y el clásico.
- Desarrollo de una aplicación para analizar la eficacia de diferentes algoritmos con una interfaz gráfica y la posibilidad de que el usuario incorpore sus propios algoritmos de forma amigable.

El paquete software realizado permite el análisis rápido de señales de ECG. Dicho paquete de software puede ayudar tanto a estudiantes como a ingenieros a realizar pruebas empíricas de sus propios algoritmos de una forma visual, ya que permite, mediante una fácil interfaz, que el usuario aplique a las señales sus propios algoritmos de análisis.

A su vez, también podría usarse para estudiantes de medicina que necesiten entrenamiento para reconocer las diferentes particularidades de la señal ECG. Para que cumpliera perfectamente esta función, ya que Matlab es un entorno orientado a la ingeniería, la aplicación debería ser reescrita en un lenguaje de programación más general.

También se han podido evaluar dos algoritmos determinando el rendimiento tanto en tasa de aciertos como en rendimiento computacional.

El primer algoritmo evaluado, desarrollado en el artículo [1] ha demostrado tener una gran tasa de aciertos. Pero requiere de varias etapas de filtrado antes de aplicar las reglas de decisión, lo que implica un mayor coste computacional.

El segundo algoritmo, desarrollado desde cero en este proyecto, consiste en una única etapa de filtrado paso banda y en calcular la transformada de Hilbert para obtener la envolvente de la salida del filtro, aplicando sobre ella unas reglas de decisión parecida a las del primer algoritmo.

Pese a presentar un menor tasa de aciertos, en las pruebas que se muestran en la tabla 4.2 se observa que este algoritmo presenta un menor coste computacional. Esta ventaja puede ser de utilidad en sistemas Holvers.

Algunas posibles líneas de ampliación de este proyecto son:

- Seguir mejorando el algoritmo desarrollado para lograr un mayor porcentaje de acierto en la detección de los complejos QRS. Para ello será conveniente evaluar el porcentaje de aciertos respecto a la longitud de la ventana que se aplica en la detección del QRS.
- Desarrollar algoritmos que permitan calcular de forma más eficiente, con menor carga computacional, la transformada de Hilbert.

- Incorporar alguna clase de interfaz que implente algún algoritmo para la detección de arritmias de forma automática.
- Incluir por defecto un algoritmo para la deducción de la frecuencia respiratoria estimada (EDR).
- Mejorar la interfaz gráfica de acuerdo a necesidades médicas.

8. Referencias Bibliográficas

- [1] Patrick S. Hamilton and **Willis J Tompkins**, “ **Quantitative Investigation of QRS Detection Rules Using the MIT/BIH Arrhythmia Database**, IEEE Transactions of biomedical engineering Vol BME 33 NO 12
- [2] Mark, R. G., HST.542J/2.792J/BE.371/6.022J “Quantitative Physiology: Organ Transport Systems”, Lecture Notes from HST/MIT Open Courseware 2004
- [3] Malmivuo and R.Plonsey, “Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields”, Oxford, U.K: Oxford University Press
- [4]H.Ito and L.Glass, “Spiral Breakup in a New Model of Discrete Excitable Media”, Phys.Rev,Lett, vol 66, No5, 1991, pp 671-674
- [5] Gari D. Clifford, Francisco Azuaje, Patrick E. McSharpy, “Advanced Methods and Tools for ECG Data Analysis”, Engineering in medicine and biology, Artech House
- [6] Joseph D Bronzino, “The Biomedical Engineering Handbook”, IEE Press
- [7] N.A.M Estes, “ECG Findings in Active Patients; Differentiating the Benign from the Serious”, The Physician and Sportsmedicine Vol 29, No3, 2001
- [8] D. di Bernardo, P.Langley and A. Murray, “Effect of Changes in Heart Rate and in Action Potential Duration On the Electrocardiogram T Wave Shape”, Physiological Measurement Vol 23, No2 2002 pp 355-364
- [9] G.F. Fletcher, “Exercise Standards: A Statement for Healthcare Professionals from the American Heart Association”, Circulation, Vol 91, No2 pp 580-615
- [10] S. Barro, “Algorithmic Sequential Decision-Making in the Frequency Domain for Life Threatening Ventricular Arrhythmias and Imitative Artefacts: A Diagnostic System”, J.Biomed. Eng No11 1989 pp 320-328
- [11] I.Jekova, A. Cansell and I.Dotsinsky, “Noise Sensivity of Three Surface ECG Fibrillation Detection Algorithms”, Physiological Measurement Vol 22 No2 2001 pp 287-297
- [12] Susan E Kelly, “Gibbs Phenomenon for Wavelets”, University of Winsconsin Lacrosse Press
- [13] J.A. Van Alsté and T.S. Schilder, “Removal of Base-Line Wander and Power-Line Interference from the ECG by an Efficient FIR Filter with a Reduced Number of Taps”, IEEE Transactions of biomedical engineering vol BME32 No12 December 1985
- [14] O.Pahlm and L.Sornmo, “Software QRS Detection in Ambulatory Monitoring – A review”, Med. Bio. Eng. Comp. Vol22 PP 289-297 1984
- [15] P.A. Lynn, “Online Digital Filters for Biological Signals: Some Fast Designs for a Small Computer”, Med. Bio. Eng. Comp. Vol15 pp 534-540

[16] N.V Thakor, J.G Webster, W.J Tompkins, “Estimation of QRS complex power spectra for design of a QRS filter” IEEE Transactions of biomedical engineering Vol BME31

[17] Alan V. Oppenheim and Ronald W. Schaffer, “Discrete-Time Signal Processing (3rd Edition)”, Prentice-Hall Signal Processing

[18] https://www.olimex.com/dev/DUINO/SHIELD-EKG-EMG/SHIELD-EKG-EMG_rev_D.pdf]

9. ANEXO: Código Fuente del Programa

Main_ecg.m:

```
function varargout = main_ecg(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @main_ecg_OpeningFcn, ...
                  'gui_OutputFcn',  @main_ecg_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before main_ecg is made visible.
function main_ecg_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to main_ecg (see VARARGIN)

% Choose default command line output for main_ecg
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
disp('Probando interfaces gráficas');
%%% valores por defecto%%%
global highpass;
global lowpass;
global notch50;
global notch60;
global fitting;
global adaptive;
highpass=1;
lowpass=1;
notch50=1;
notch60=0;
fitting=0;
adaptive=0;
```

```

%%

% UIWAIT makes main_ecg wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = main_ecg_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} =handles.output;

% --- Executes on button press in botonejecutar.
function botonejecutar_Callback(hObject, eventdata,handles)
% hObject handle to botonejecutar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global archivo;
global ruta;
global ecgvar;
global tsamp;

%%
global highpass;
global lowpass;
global notch50;
global notch60;
global fitting;
global adaptive;
%%

%%
global rr_interval;
global latidos;
global R_value;
global width;
%%

%%
global tini;
global tfin;
%%

%% Inicialización algunas variables
tsamp=0.0028; % frecuencia de muestreo 360Hz
time_ecgvar = (length(ecgvar))*tsamp
tini = NaN;
tfin = NaN;

%% Eleccion de filtros Baseline y 60 Hz%%

```

```

cd C:\proyecto\funciones\filters
highpass
lowpass
notch50
notch60
fitting
adaptive

if highpass==1
    filtro_hp=filters(1);
    ecgvar=baseline_removal(ecgvar,filtro_hp);
    %assignin('base','bu',ecgvar);
end

if lowpass==1
    filtro_lp=filters(2);
    baseline=baseline_removal(ecgvar,filtro_lp);
end

if notch50==1
    filtro_notch50=filters(3);
    ecgvar=baseline_removal(ecgvar,filtro_notch50);
end

if notch60==1
    filtro_notch60=filters(4);
    ecgvar=baseline_removal(ecgvar,filtro_notch60);
end

if fitting==1
    % por desarrollar
end

if adaptive==1
    [ecgvar,error]=adaptive(x,tsamp);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

cd(ruta) % excepcion para errores
if ecgvar ~= 0
    tic
    [R_value,points]=QRS_detector(ecgvar,tsamp);
    timeCPU = toc

    cd C:\proyecto\funciones
    %%% Heart Rate%%
    [latidos,rr_interval]=beat_counter(points,tsamp);

    %%%

    %%% QRS width%%
    % La función QRS_width solo es fiable si previamente se ha
    eliminado el
    % baseline wander, en el caso de no detectar dicha operacion, la
    funcion no
    % se ejecuta.

```

```

if highpass==1 || fitting==1
    [puntos,width]=qrs_width(ecgvar,points,tsamp);

else
    no_baseline_removal

    for i=0:length(points)
        width(i)=0;
    end
end

end

%%%

%%% generacion Logs%%%
[filename,pathfile]=log_generator(points,tsamp,R_value,rr_interval,lat
idos,width);
%%%

tini = get(handles.tini,'String');
tfin = get(handles.tfin,'String');

tini = cell2mat(tini);
tfin = cell2mat(tfin);

tini = str2num(tini);
tfin = str2num(tfin);

if tini < time_ecgvar && (tfin <= time_ecgvar || tfin < tini)

    sampleIni = round(tini/tsamp);
    sampleFin = round(tfin/tsamp);

    realTimeIni = sampleIni*tsamp;
    realTimeFin = sampleFin*tsamp;

    n=1;
    tag = false;

    while points(n) <= sampleFin

        if points(n) >= sampleIni && tag == false

            initialPoint = n;
            tag = true;

        end

        n = n+1;
    end
end

```

```

finalPoint = n-1;

time=realTimeIni:tsamp:realTimeFin;

xMin = realTimeIni;
xMax =realTimeFin;
yMin = -2;
yMax = 3;

descripcion='Pruebas QRS detector( Esto sera un output obligatorio
en la definicion de la funcion';
set(handles.edit5,'string',descripcion);

handles.ECG_screen=plot(time,ecgvar(sampleIni:sampleFin),(points(initialPoint:finalPoint))*tsamp,R_value(initialPoint:finalPoint)); %
presionando el boton muestra en el grafico

axis([xMin xMax yMin yMax]);

else

%%% DialogBox mostrando ERROR en los valores

end

%%%

cd C:\proyecto\funciones
%plot(ECG_screen,y);

else

%%%%% DIALOGBOX informando de que no hay señal ECG
cargada(Pendiente)
end

% --- Executes on key press with focus on botonejecutar and none of
its controls.
function botonejecutar_KeyPressFcn(hObject, eventdata, handles)

```

```

% hObject      handle to botonejecutar (see GCBO)
% eventdata    structure with the following fields (see UIControl)
%   Key: name of the key that was pressed, in lower case
%   Character: character interpretation of the key(s) that was pressed
%   Modifier: name(s) of the modifier key(s) (i.e., control, shift)
pressed
% handles      structure with handles and user data (see GUIDATA)

function edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of edit2
as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function File_Callback(hObject, eventdata, handles)
% hObject      handle to File (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function Load_Callback(hObject, eventdata, handles)
% hObject      handle to Load (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global ecgvar;
[file,pathname]=uigetfile({'*.mat'},'Open File');
if file ~= 0

    cd (pathname);
    ecgvar=open (file);
    ecgvar=ecgvar.val(1,:);
    cd C:\proyecto\funciones
    ecgvar=raw2phy(ecgvar,1024,200);
else

```

```

        ecgvar = 0;

end

% --- Executes on button press in clear.
function clear_Callback(hObject, eventdata, handles)
% hObject      handle to clear (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
plot(0);

% -----
function LoadQRS_Callback(hObject, eventdata, handles)
% hObject      handle to LoadQRS (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
[file,pathname]=uigetfile({'*.m','Open File'});

global archivo;
global ruta;

archivo=file;
ruta=pathname;

function signal_param_Callback(hObject, eventdata, handles)
% hObject      handle to signal_param (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of signal_param as
text
%         str2double(get(hObject,'String')) returns contents of
signal_param as a double

% --- Executes during object creation, after setting all properties.
function signal_param_CreateFcn(hObject, eventdata, handles)
% hObject      handle to signal_param (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject      handle to edit5 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5
%        as a double

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function close_Callback(hObject, eventdata, handles)
% hObject      handle to close (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
close main_ecg;

% -----
function filter_Callback(hObject, eventdata, handles)
% hObject      handle to filter (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function options_Callback(hObject, eventdata, handles)
% hObject      handle to options (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
filtering_options

% -----
function data_analysis_Callback(hObject, eventdata, handles)
% hObject      handle to data_analysis (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function heart_rate_Callback(hObject, eventdata, handles)
% hObject      handle to heart_rate (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```


heart_rate

```
% -----
function r_wave_Callback(hObject, eventdata, handles)
% hObject    handle to r_wave (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
r_wave

% -----
function qrs_width_Callback(hObject, eventdata, handles)
% hObject    handle to qrs_width (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function custom_Callback(hObject, eventdata, handles)
% hObject    handle to custom (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function hrv_custom_Callback(hObject, eventdata, handles)
% hObject    handle to hrv_custom (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

custom

% -----
function rwave_custom_Callback(hObject, eventdata, handles)
% hObject    handle to rwave_custom (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

custom % usar señal R_wave

% -----
function qrs_custom_Callback(hObject, eventdata, handles)
% hObject    handle to qrs_custom (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

custom % usar señal QRS_width

function tini_Callback(hObject, eventdata, handles)
% hObject    handle to tini (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tini as text
%        str2double(get(hObject,'String')) returns contents of tini as
%        a double
```

```

% --- Executes during object creation, after setting all properties.
function tini_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tini (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tfin_Callback(hObject, eventdata, handles)
% hObject    handle to tfin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tfin as text
%        str2double(get(hObject,'String')) returns contents of tfin as
a double

% --- Executes during object creation, after setting all properties.
function tfin_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tfin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

Filters.m:

```
function[filtro]=filters(opcion)

% Funcion que inicializa filtros para eliminar baseline Wander y
% obtener
% dicho baseline por separado.
% Filtro Paso alto fc=0.8 Hz
% Con la variable opcion eliges que filtro inicializar

if opcion==1

Fs = 360; % Sampling Frequency

Fstop = 0.6; % Stopband Frequency
Fpass = 0.8; % Passband Frequency
Dstop = 0.0177827941; % Stopband Attenuation
Dpass = 0.057501127785; % Passband Ripple
dens = 20; % Density Factor

% Calculate the order from the parameters using FIRPMORD.
[N, Fo, Ao, W] = firpmord([Fstop, Fpass]/(Fs/2), [0 1], [Dstop,
Dpass]);

% Calculate the coefficients using the FIRPM function.
filtro = firpm(N, Fo, Ao, W, {dens});

end

if opcion==2

% Filtro Paso bajo fc=0.8 Hz 1 Hz

Fs = 360; % Sampling Frequency

Fpass = 0.6; % Passband Frequency
Fstop = 0.8; % Stopband Frequency
Dpass = 0.057501127785; % Passband Ripple
Dstop = 0.0177827941; % Stopband Attenuation
dens = 20; % Density Factor

% Calculate the order from the parameters using FIRPMORD.
[N, Fo, Ao, W] = firpmord([Fpass, Fstop]/(Fs/2), [1 0], [Dpass,
Dstop]);

% Calculate the coefficients using the FIRPM function.
filtro = firpm(N, Fo, Ao, W, {dens});
```

```

end

if opcion==3
% Filtro notch 50 Hz

Fs = 360; % Sampling Frequency

Fpass1 = 48; % First Passband Frequency
Fstop1 = 49; % First Stopband Frequency
Fstop2 = 51; % Second Stopband Frequency
Fpass2 = 52; % Second Passband Frequency
Dpass1 = 0.028774368332; % First Passband Ripple
Dstop = 0.01; % Stopband Attenuation
Dpass2 = 0.028774368332; % Second Passband Ripple
dens = 20; % Density Factor

% Calculate the order from the parameters using FIRPMORD.
[N, Fo, Ao, W] = firpmord([Fpass1 Fstop1 Fstop2 Fpass2]/(Fs/2), [1 0
...
1], [Dpass1 Dstop Dpass2]);

% Calculate the coefficients using the FIRPM function.
filtro = firpm(N, Fo, Ao, W, {dens});
end

if opcion==4
% Filtro notch 60 Hz

Fs = 360; % Sampling Frequency

Fpass1 = 58; % First Passband Frequency
Fstop1 = 59; % First Stopband Frequency
Fstop2 = 61; % Second Stopband Frequency
Fpass2 = 62; % Second Passband Frequency
Dpass1 = 0.057501127785; % First Passband Ripple
Dstop = 0.01; % Stopband Attenuation
Dpass2 = 0.028774368332; % Second Passband Ripple
dens = 20; % Density Factor

% Calculate the order from the parameters using FIRPMORD.
[N, Fo, Ao, W] = firpmord([Fpass1 Fstop1 Fstop2 Fpass2]/(Fs/2), [1 0
...
1], [Dpass1 Dstop Dpass2]);

% Calculate the coefficients using the FIRPM function.
filtro = firpm(N, Fo, Ao, W, {dens});
end

```

Filtering.m:

```
function [z,yfpa]=filtering(x,ts)
% x señal de entrada ECG
% ts tiempo de muestreo de la señal
% y señal de salida ( para pruebas )
% tm matriz tiempos deteccion QRS

% Normalizacion entrada.
norm=max(x);
x=x/norm;
longitud_signal=length(x);

%Definimos filtro paso bajo
a=[1,-2,1]; %y[n]
b=[1,0,0,0,0,0,-2,0,0,0,0,0,1]; %x[n]
n=(0:ts:length(x)-1); %dudas con ese ts

%Definimos filtro paso alto
c=[1,-1]; %y[n]
d=[-1/32,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,-
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1/32]; %x[n]

%Ecuacion en diferencias del diferenciador
e=[1]; %y[n]
f=[2/8,1/8,0,-1/8,-2/8]; %x[n]

%Etapa filtro paso bajo
yfpb=filter(b,a,x);

%Etapa filtro paso alto
yfpa=filter(d,c,yfpb);

%Etapa diferenciador
yfd=filter(f,e,yfpa); % fiducial mark locator

%Non linear squaring function
ynf=yfd.^2;

%Time averaged window%
z(1)=sum(ynf(1:32))/32;
i=1;
j=2;
while (j*32+1)< longitud_signal
    z(j)=sum(ynf(i*32+1:j*32+1))/32;
    i=i+1;
    j=j+1;
end
```

Peakdetector2.m:

```
function [y] = peak_detector2(x)
%PEAK_DETECTOR detecta los puntos donde existe un máximo
% x Input vector
% y Maximum points
% longitud de la muestra
derivative=diff(x);
long=length(derivative);
j=1;
n=1;

marcador=0;

while n<long % n hay que tocarlo manualmente, con lo cual es mejor un
while
    if derivative(n)>0.5 %2

        [valor,punto]= max(x(n-1:n+2)); %[n-1, n, n+1, n-2]

        if punto==1
            punto=n-1;

        end

        if punto==2
            punto=n;
        end

        if punto==3
            punto=n+1;
        end

        if punto==4
            punto=n+2;
        end

        end

        y(j,1)=punto;
        y(j,2)=x(punto);
        j=j+1;
        marcador=1;
    end

    if marcador==0
        n=n+1;
    else
        n=n+3;
    end
    end
    marcador=0;
```

end

Qrs_rule.m:

```
function [fiducial,posiciones]=qrs_rule(x,maximos)
% Fiducial= vector con valores maximos en la señal time averaged
% posiciones= posiciones punto QRS
% Points= puntos en la señal original donde se localiza el QRS
% Time= diferencia temporal entre complejos QRS
% x= señal original
% Tsamp= periodo de muestreo
maximos=maximos(:,1);
maximos=maximos';
long_max=length(maximos);

for i=1:long_max

    posicion_high=32*maximos(i); % no definitivo
    posicion_low=posicion_high-64; %posiciones definidas por el maximo

    [fiducial(i),posiciones(i)]=max(x(posicion_low:posicion_high));

    if posiciones(i)==1
        posiciones(i)=posicion_low;
    end

    if posiciones(i)>1
        posiciones(i)=posicion_low+(posiciones(i)-1);
    end

end
```

End

QRS_detector.m:

```
function [R_value,points]=QRS_detector(ecg,tsamp)

% Prototipo de funcion estandar para deteccion QRS
% ECG=Señal ECG
% tsamp=tiempo de muestreo
% R_value(i)= Valores de pico de las ondas R
% Points(i)= puntos del vector ECG donde se encuentra la onda R

[ysal,yfp]=filtering(ecg,tsamp);
maximos=peak_detector2(ysal);
[R_value,points]=qrs_rule(ecg,maximos);
```


QRS_detector (algoritmo2):

```
function [R_value,points]=QRS_detector(ecg,tsamp)

%%% Creacion filtro paso banda fc= 17 Hz %%%%%%%%%%%%%%
Fs = 1/tsamp; % Sampling Frequency

Fstop1 = 15; % First Stopband Frequency
Fpass1 = 16; % First Passband Frequency
Fpass2 = 18; % Second Passband Frequency
Fstop2 = 19; % Second Stopband Frequency
Dstop1 = 0.001; % First Stopband Attenuation
Dpass = 0.028774368332; % Passband Ripple
Dstop2 = 0.001; % Second Stopband Attenuation
dens = 20; % Density Factor

% Calculate the order from the parameters using FIRPMORD.
[N, Fo, Ao, W] = firpmord([Fstop1 Fpass1 Fpass2 Fstop2]/(Fs/2), [0 1
...
                                0], [Dstop1 Dpass Dstop2]);

% Calculate the coefficients using the FIRPM function.
b = firpm(N, Fo, Ao, W, {dens});
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tic
y=baseline_removal(ecg,b);
z=abs(hilbert(y));
assignin('base','bu',z);

%points = peak_detector2(z);

zDiff = diff(z);
i=1;
position = 26;
n = 40;
isPositive = false;
isNegative = false;

while n <= length(zDiff)

    if zDiff(n) > 0

        isPositive = true;

    else

        isNegative = true;
```

```

end

if isPositive == true && isNegative == true

    if n+25 < length(ecg)

        subEcg = ecg(n-25:n+25); % 10*0.0028 = 0.0800 s
        [value,index] = max(subEcg);

        if value >= 0.6

            if position < index

                shift = position - index;
                realIndex = n - shift;

            else

                shift =abs(position - index);
                realIndex = n + shift;

            end

            R_value(i) = value;
            points(i) = realIndex;

            isPositive = false;
            isNegative = false;

            i = i+1;

            n= n+25;
        else

            isPositive = false;
            isNegative = false;

        end

    else

        isPositive = false;
        isNegative = false;

    end

end

end
n = n+1;
end

```

```
% assignin('base','bupoints',points);
timePU = toc

end
```

Log_generator.m:

```
function
[filename,pathfile]=log_generator(posiciones,tsamp,fiducial,rr_interva
l,heart_beats,qrs_width)
% Filename= Nombre del log
% Pathname= Ruta del log
% Posiciones= Posiciones QRS
% tsamp= Tiempo de muestreo
% fiducial= Valores de pico onda R
% rr_interval= Intervalo rr
% heart_beats= latidos

%Nombre de archivo y creacion de fichero
day=date;
pathfile= 'C:\proyecto\funciones\Logs'; % no definitivo
cd (pathfile);
filename=strcat(date,'-', 'log.txt');
file=fopen(filename,'w');

%Quizas otros posibles datos ( duracion señal,tiempo de muestreo...

%Escribir encabezados
fprintf(file,'Time (s)\t');
fprintf(file,'R wave (V)\t');
fprintf(file,'QRS (s)\t');
fprintf(file,'RR interval (s)\t');
fprintf(file,'Heart Rate (bpm)\n');

%Escribir datos señal ECG
long_posiciones=length(posiciones); % igual a la longitud de fiducial

for i=1:long_posiciones
    tiempo=posiciones(i)*tsamp;
    fprintf(file,'%f\t',tiempo);
    fprintf(file,'%f\t',fiducial(i));
    fprintf(file,'%f\t',qrs_width(i));

    if i<long_posiciones
        fprintf(file,'%f\t',rr_interval(i));
        fprintf(file,'%f\n',heart_beats(i));
    end

end

%Cerrar archivo
cd C:\proyecto\funciones
fclose(file);
```

Qrs_width:

```
function [points,width]=qrs_width(x,r_points,tsamp)
% Points= punto extremo izq y dcho de un QRS
% width= anchos del complejo QRS
% x= señal ecg de entrada
% r_points= puntos onda R señal de entrada
long_r=length(r_points);
longEcg=length(x);

for i=1:long_r

    if (r_points(i)+36) < longEcg
        vector=x(r_points(i)-36:r_points(i)+36); %72*0.0028=0.20s

        %[valor,posicion]=max(vector);
        posicion =37;
        desplazamiento=1;
        contador_izq=0;
        contador_dcha=0;

        while (posicion-(desplazamiento)) > 0 && vector(posicion-
(desplazamiento-1))>vector(posicion-desplazamiento)
            contador_izq=contador_izq+1;
            desplazamiento=desplazamiento+1;
        end
        contador_izq=contador_izq+1;

        %Baseline
        while (posicion-contador_izq)>0 && vector(posicion-contador_izq)<-
0.05
            contador_izq=contador_izq+1;
        end

        desplazamiento=1;

        while (posicion+desplazamiento) > longEcg &&
vector(posicion+(desplazamiento-1))>vector(posicion+desplazamiento)

            contador_dcha=contador_dcha+1;
            desplazamiento=desplazamiento+1;
        end
        contador_dcha=contador_dcha+1;

        %baseline

        while (posicion+contador_dcha)<length(vector) &&
vector(posicion+contador_dcha)<-0.05
            contador_dcha=contador_dcha+1;
```

```

end

points(i)=contador_izq+contador_dcha;
width(i)=points(i)*tsamp;

end
Adaptive.m:

function [y,error,desired]=adaptive(x,tsamp)
% 50Hz Power line interference

t=0:tsamp:(length(x)-1)*tsamp;
desired=sin(2*pi*50*t);
length(desired)
h=adaptfilt.lms(30,0.002);
[y,error]=filter(h,x,desired);

y=x-y;

```